

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-170656

(43)Date of publication of application : 29.06.1999

(51)Int.Cl.

B41J 21/00
B41J 5/30

(21)Application number : 09-343223

(71)Applicant : CANON INC

(22)Date of filing : 12.12.1997

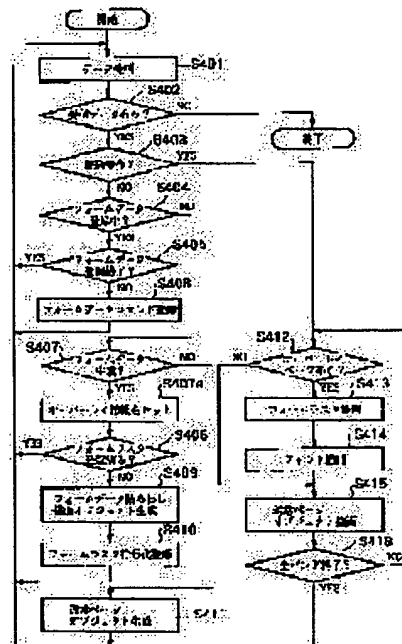
(72)Inventor : TSUNEKAWA KIYOHIRO

(54) METHOD AND APPARATUS FOR OUTPUTTING IMAGE

(57)Abstract:

PROBLEM TO BE SOLVED: To perform plotting process at high speed regardless of the content of a form data.

SOLUTION: When a print data is received (S401), instruction thereof is determined. If it is an instruction for generating a form data, a form data is generated as a raster data and stored while being encoded except the character data contained therein (S410). If it is a print instruction, a decision is made whether it is an overlay page (S412) and if the answer is YES, the encoded form data is decoded and written and then a data to be printed is developed thereon (S413) whereas the character in the form data is read out from a font cache and written before being outputted. Character in the form data may be rendered not to be deleted easily from the font cache.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] A form storing means to encode and store the form data generated as bit map data which do not contain an alphabetic character, A form generation means to generate form data as bit map data, to encode, and to store in said form storing means, The font cache which stores the alphabetic data generated as bit map data, An alphabetic character generation means to generate alphabetic data as bit map data, and to store in said font cache, In case print data are overlaid with form data and printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is an image output unit characterized by having an overlay means to draw using the bit map data stored in said font cache, and to output said print data in piles.

[Claim 2] Said form generation means is an image output unit according to claim 1 characterized by generating raster data, encoding and storing based on the form data received with print data.

[Claim 3] Said form generation means is an image output unit according to claim 1 characterized by generating raster data, encoding and storing based on the form data beforehand stored in nonvolatile memory.

[Claim 4] Said form generation means is an image output unit according to claim 3 characterized by reading form data to a power up or idle time, and generating a form image at it.

[Claim 5] Said form generation means is an image output unit according to claim 3 characterized by generating a form image with the form generation instruction directed by the host.

[Claim 6] Said form generation means is an image output unit according to claim 1 characterized by generating raster data, encoding and storing in the band unit which comes to divide 1 page.

[Claim 7] Said form generation means is an image output unit according to claim 1 characterized by generating raster data per 1 page, encoding, and storing.

[Claim 8] Said overlay output means is an image output unit according to claim 1 characterized by overlapping usual page data other than form data on form data.

[Claim 9] Said alphabetic character generation means is an image output unit according to claim 1 characterized by generating the alphabetic data containing the alphabetic data contained in form data as an image data, and storing in said font cache.

[Claim 10] Said alphabetic character generation means is an image output unit according to claim 1 characterized by deleting from the low alphabetic data of priority in deleting the data stored in the font cache, in case priority higher than the alphabetic data usually contained in page data is set up, it stores in a font cache and alphabetic data is developed to a bit map about the alphabetic data contained in form data.

[Claim 11] The form generation process which generates the form data which do not contain an alphabetic character as bit map data, encodes, and is stored in a form storing means, The alphabetic character generation process which generates alphabetic data as bit map data, and is stored in a font cache, in case it overlays print data with form data and they are printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is an image output method characterized by having the overlay process which draws using the bit map data stored in said font cache, and outputs said print data in piles.

[Claim 12] Said form generation process is an image output method according to claim 11 characterized by generating raster data, encoding and storing based on the form data received with print data.

[Claim 13] Said form generation process is an image output method according to claim 12 characterized by generating raster data, encoding and storing based on the form data beforehand stored in nonvolatile memory.

[Claim 14] Said form generation process is an image output method according to claim 13 characterized by reading form data to a power up or idle time, and generating a form image at it.

[Claim 15] Said form generation process is an image output method according to claim 13 characterized by generating a form image with the form generation instruction directed by the host.

[Claim 16] Said form generation process is an image output method according to claim 11 characterized by generating raster data, encoding and storing in the band unit which comes to divide 1 page.

[Claim 17] Said form generation process is an image output method according to claim 11 characterized by generating raster data per 1 page, encoding, and storing.

[Claim 18] Said overlay output process is an image output method according to claim 11 characterized by overlapping usual page data other than form data on form data.

[Claim 19] Said alphabetic character generation process is an image output method according to claim 11 characterized by deleting from the low alphabetic data of priority in deleting the data stored in the font cache, in case priority higher than the alphabetic data usually contained in page data is set up, it stores in a font cache and alphabetic data is developed to a bit map about the alphabetic data contained in form data.

[Claim 20] A form generation means to generate the form data which do not contain an alphabetic character as raster data, to encode, and to store in a form storing means, An alphabetic character generation means to generate alphabetic data as bit map data, and to store in a font cache, in case it overlays print data with form data and they are printed, while decoding the stored form data which were encoded The alphabetic data contained in form data draws using the bit map data stored in said font cache. Computer-readable memory characterized by storing the program operated as a computer as equipment which has an overlay means to output said print data in piles.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to printers, such as the image output unit and approach in which an overlay output of page data and form data is for example usually possible, especially a laser beam printer.

[0002]

[Description of the Prior Art] Atypical data divisions in which the contents differ from the fixed form form sections (a title, closing line, etc.) which print the same contents over two or more pages for every page like a document or a fixed form document in printing on general business (a figure, identifier, etc.) usually, page data — calling — form overlay printing piled up and printed is used in many cases.

[0003] Then, in the conventional page printer which analyzes a Page Description Language and generates a raster image, it registered with the storage region of a printer by making into event drive macro instruction the fixed form form section created in said Page Description Language format, and it is carrying out automatic activation of said macro by events, such as delivery actuation, and form overlay printing was realized. The time of assignment of this macro (1st activation), and after the analysis of the usual page data for 1 page is completed, whenever it receives a delivery instruction (2nd henceforth), the contents of registration are read, and said event drive macro is usually analyzed like page data by the Page Description Language analysis means, and is changed into the pseudo code (it is hereafter called a drawing object) for generating the raster image for 1 page.

[0004] Moreover, it may be constituted so that said form data (macro data) may be beforehand transmitted to a printer from a host computer and it is not necessary to transmit form data by registering to nonvolatile memory, such as a flash memory, each time in advance of printing activation. In this case, only the instruction which directs the form data in a flash memory as an event drive macro will be sent at the time of printing activation.

[0005]

[The technical problem which invention will solve and to carry out] However, there was a trouble that it explained below, in the above-mentioned conventional example.

[0006] That is, since form data were registered in a Page Description Language format when said event drive macro has realized form overlay printing, in the case of complicated form data, the analysis of the form data itself and generation of a drawing object had taken time amount. Namely, the more only processing speed equivalent to the case where the output of the 1st page usually outputs the contents same as data, without using overlay was obtained and the contents of form became complicated, the more the processing time of the form section was increasing.

[0007] Moreover, in case the print data which are using the same form by two or more pages are outputted, in order to analyze the same macro (command group) for every page each time and to generate the same drawing object, in spite of having been the page of the same contents, there was a trouble that the printing speed after the 2nd page did not improve. That is, since ** page generation was carried out, sharing of a drawing object was not able to do the same drawing object in two or more pages. Therefore, memory required for two or more pages drawing processing increased, and we were anxious also about the performance fall of the waiting for delivery

occurring. When output specification of the overlay is carried out, the error of drawing insufficient memory etc. also becomes easy to generate the complicated form which contains many graphic forms and image images especially, and solution of the above-mentioned technical problem was important.

[0008] As one of the solutions of the above-mentioned problem, form data are rasterized beforehand, it holds in the printer as an image data (form image), and it is possible to usually pile up with page data and to draw. However, in order to hold the image for 1 page, a lot of memory is needed, and it is not practical (it is A4, 600dpi, and monochrome and about 8MB of memory is the need).

[0009] Then, although it was possible to agreement-ize a form image (compression), since the compressibility of the pattern containing an alphabetic character generally was not good, the pack BITTSU compression (an image is agreement-ized paying attention to the continuity of a byte order) which can be adapted for compression of a form image had the problem that the compressibility of a form image will fall, when many alphabetic data was contained in form data.

[0010] Even if it was made in view of the above-mentioned trouble of this invention and the contents of form become complicated, the processing time of the form section is not increased, and if it is the form of the same contents, the printing speed after the 2nd page can be raised, and memory required for the drawing processing which is two or more pages is not increased, and it aims at offering the ***** output unit and approach of not reducing performance, such as waiting for delivery.

[0011]

[Means for Solving the Problem] This invention for solving the above-mentioned trouble is equipped with the following configurations. Namely, a form storing means to encode and store the form data generated as bit map data which do not contain an alphabetic character, A form generation means to generate form data as bit map data, to encode, and to store in said form storing means, The font cache which stores the alphabetic data generated as bit map data, An alphabetic character generation means to generate alphabetic data as bit map data, and to store in said font cache, In case print data are overlaid with form data and printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is an image output unit equipped with an overlay means to draw using the bit map data stored in said font cache, and to output said print data in piles.

[0012] Or the form generation process which generates the form data which do not contain an alphabetic character as bit map data, encodes, and is stored in a form storing means, The alphabetic character generation process which generates alphabetic data as bit map data, and is stored in a font cache, in case it overlays print data with form data and they are printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is an image output method equipped with the overlay process which draws using the bit map data stored in said font cache, and outputs said print data in piles.

[0013] Or a form generation means to generate the form data which do not contain an alphabetic character as raster data, to encode, and to store in a form storing means, An alphabetic character generation means to generate alphabetic data as bit map data, and to store in a font cache, in case it overlays print data with form data and they are printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is the computer-readable memory which stored the program operated as a computer as equipment which has an overlay means to draw using the bit map data stored in said font cache, and to output said print data in piles.

[0014]

[Embodiment of the Invention] Before explaining the configuration of this operation gestalt, the configuration of a suitable image processing system to apply this invention is explained referring to drawing 1 . Drawing 1 is the sectional view showing the configuration of the image processing system which can apply this invention, for example, shows the case of a laser beam printer (henceforth, LBP). In addition, the image processing system which can apply this invention is not restricted to LBP, and it cannot be overemphasized that you may be printer equipment of other print methods.

[0015] in this drawing , the LBP body 100 create a character pattern , a graphic form pattern , etc.

which correspond according to those information, and form an image in the record paper which be a record medium while it input and memorize the text and graphic forms (character code etc.) information which be supply in the form of a Page Description Language, fixed form format data (form data) or other macro instruction, etc. from the host computer (200 of drawing 2) connect outside. As for the control panel 112, a switch, an LED drop, etc. for actuation are allotted. The printer control unit 101 analyzes the text supplied from control and the host computer of the LBP100 whole. Text is changed into the corresponding video signal of a character pattern, and this control unit 101 mainly outputs it to a laser driver 102. Moreover, this LBP can also perform registration of registration of a character pattern, form data, etc. from the non-illustrated source of data, is similarly changed into a video signal according to printing data, and is outputted to a laser driver 102. A laser driver 102 is a circuit for driving semiconductor laser 103, and carries out the on-off change of the laser beam 104 discharged from semiconductor laser 103 according to the inputted video signal. Laser 104 is shaken at a longitudinal direction by the rotating polygon 105, and scans the electrostatic drum 106 top. Thereby, the electrostatic latent image of a character pattern is formed on the electrostatic drum 106. After this latent image is developed by the development unit 107 of electrostatic drum 106 perimeter, it is transmitted to the recording paper. - uses a cut sheet for this recording paper, and the cut sheet recording paper is contained by the form cassette 108 with which LBP100 was equipped, is incorporated in equipment with the feed roller 109 and the conveyance rollers 110 and 111, and is supplied to the electrostatic drum 106.

[0016] With reference to an attached drawing, the gestalt of operation of the 1st of this invention is explained below to [the gestalt of the 1st operation].

[0017] Drawing 2 is a block diagram explaining the basic configuration (internal configuration of the control unit 101 of the image processing system of drawing 1) of the image output unit in which the gestalt of operation of the 1st of this invention is shown, for example, shows the case of a laser beam printer.

[0018] Moreover, as long as the function of this invention is performed, even if it is the device of a simple substance, you may be the system by which processing is performed through networks, such as LAN.

[0019] In this drawing, it connects with the external host computer 200 through the predetermined interface 208, and the body 100 of a printer is constituted so that predetermined printing processing may be performed according to the print data sent from a host computer 200. These print data received from the host computer 200 are temporarily stored in a receive buffer 207 as received data.

[0020] In addition, the form data described in the Page Description Language format are stored in the hard disk of a host computer 200, and each form data are matched with the information on the identifiable form ID etc., and are managed by the meaning. Moreover, said form data are sent out as a form registration instruction (event drive macro) at the time of form overlay printing.

[0021] A printer CPU 201 controls in generalization access with the various devices connected to the system bus 206 based on the control program 203 memorized by ROM202 which is a read-only memory. Analyze the print data stored in the receive buffer 207, and the drawing object for at least 1 page is generated (drawing object generation section 203a). Then, the raster image for one band divided into 1 page or two or more bands based on said drawing object is generated (drawing object drawing section 203c). The video signal by ON/OFF of the bit of said raster image is outputted to the printing section 210 (printer engine) through printing section I/F and 209 (cyripedium is carried out). Performing the printing section 210 based on the video signal received from CPU201, it prints an image on the recording paper 211. Here, the font scaler which generates the font pattern corresponding to alphabetic data (a typeface, size, character code) is also contained in drawing object generation section 203a.

[0022] In addition, when preparing a part for at least 2 bands which divided the page in the shape of a band, without preparing the memory which stores a raster image by 1 page (henceforth, band raster), it is constituted so that generation and cyripedium actuation of said raster image may be performed to coincidence. That is, the drawing object belonging to the band raster of another side is drawn, carrying out the cyripedium of one band raster [finishing / drawing] (the above-mentioned drawing procedure is hereafter called drawing by banding).

[0023] Moreover, the control program 203 consists of programs for realizing the following functions in addition to the above-mentioned configuration.

[0024] Namely, when the print data in a receive buffer 207 are form registration instructions The overlay registration / read-out section 203b which registers said form data described by the Page Description Language into RAM205 (after-mentioned), and reads said form data at the time of form overlay run command reception, Drawing object drawing section 203c to which the form raster which does not contain an alphabetic character based on the read form data is generated, and alphabetic data generates a font pattern, 203d of form memory management sections which memorize and hold said form raster in RAM205, 203g of font cache Management Department which reuses a registered font pattern, without using a font scaler when said font pattern is registered to the font cache storing section and the output of the same alphabetic data is required, 203f of form compression / expanding sections decrypted before agreement-izing before memorizing said form raster, usually piling up with a page and drawing, It consists of form raster drawing section 203e which usually piles up a form raster [finishing / said decryption] with page data, and draws to page memory 205c.

[0025] In addition, the form raster corresponding to the demanded form data investigates whether it is finishing [storage in 205d of form raster storing sections], and 203d of said form memory management sections is equipped with the cache function to use a memorized form raster when it is already storage ending (however, in order to use a font cache [finishing / generation] about the alphabetic data in form data, read-out of form data is performed from overlay storing section 203a). Moreover, this ROM202 is used also as memory which stores the font data which consists of a dot font used for an alphabetic character output, and a scalable font (font ROM 204).

[0026] RAM205 is RAM which functions as the main memory of CPU201, work-piece memory, etc., and it is constituted so that memory space can be extended with the option RAM connected to the extension port which is not illustrated. Drawing object storing section 205a (the font cache registration memory which carries out the cache of the character pattern developed based on the scalable data in said font memory 204 is included) in which RAM205 stores said drawing object, Work-piece memory 205e temporarily used by control program 203a, Page memory 205c which stores the raster image (or raster image for 1 page) equivalent to two regions of the band field which divided 1 page in the shape of a band, It is used for printing environmental data storage memory besides 205d of form raster storing sections for agreement-izing the form image which analyzed form data storage section 205b which stores the form data described by the Page Description Language, and said form data, and was obtained, and holding it etc. (un-illustrating). In addition, LBP100 has received supply of power from the power supply section which does not illustrate. Moreover, although it explained that form data were stored in RAM205 above, you may store in nonvolatile memory, such as a flash memory, beforehand.

[0027] In addition, although it explained that the equipment which constitutes this invention was supplied as a program stored in ROM, after being supplied by media, such as not only this but a floppy disk, and a hard disk, and being loaded to RAM etc. before activation, it may be constituted so that it may perform.

The procedure of the overlay output in a <overlay output-processing procedure>, then the image output unit by the above-mentioned configuration is explained below using the flow chart of drawing 3 and drawing 4 .

[0028] When <outline of form data registration> drawing 3 (A) receives form data from a host computer 200, it is the flow chart which showed the outline of the instruction procedure for generating and registering a form raster.

[0029] First, in step S301, if a form registration instruction is received, form data will be registered to form data storage section 205b in a printer. Here, a form registration instruction means a series of instruction groups which consist of live data of an overlay registration initiation instruction - form - an overlay registration termination instruction. Then, according to a form lath evening generation instruction, the form data registered at step S301 are read (step S302), and the drawing object which analyzes form data and contains the corresponding font pattern for 1 page is generated (step S303). In addition, the generated font pattern is usually registered to the font cache like the alphabetic data of a page.

[0030] Then, drawing objects other than a font object are drawn to a band raster, and a form

raster is generated (step S304). If generation of a form raster is completed, after agreement-izing said form raster at step S305, a form raster is registered with the information on the form ID for identifying form etc., and form raster registration processing is ended.

[0031] <Outline of form overlay activation> drawing 3 (B) is the flow chart which showed the outline of the procedure at the time of receiving a form overlay run command. In addition, reception and processing of registration of the form data shown in drawing 3 (A) and form overlay activation of drawing 3 (B) may be done continuously (with the same print job), and it does not matter even if it performs only registration of form data previously.

[0032] In this drawing, it investigates first whether the form raster corresponding to the form ID specified by the form overlay run command is registered in step S310 (203d of form raster Management Department). When already registered, the directions which draw this form raster are told to form raster drawing section 203e (step S312), only a part for the alphabetic character data division of form data is analyzed to coincidence, and a font object is generated (step S313). Since a font object hits to the font cache registered at step S303 of drawing 3 (A) at this time, it is not carried out by the scaling processing by the scaler.

[0033] On the other hand, when judged [having not registered a desired form raster at step S310, and], at step S311, all form data are read and a drawing object is generated. What is necessary is just to perform the generation of a drawing object itself like the time of form raster registration.

[0034] After processing of form data is completed as mentioned above, the usual page data of the atypical section are analyzed and a drawing object is generated like step S311 etc. (S314).

[0035] After usually processing page data with the form data for 1 page, band RASUTAHE drawing of the form part is carried out (step S315). That is, the font object contained in an agreement-ized form raster and form when a form raster is registered is drawn, and on the other hand, when having not registered, the drawing object group (a font object is included) equivalent to a form part is drawn to a band raster. Finally, the drawing object of a page is usually drawn similarly (step S316), and the executive operation of a form overlay is ended.

[0036] Now, although the flow chart shown in drawing 3 shows independently the procedure which shows registration of form, and the procedure which shows activation of a form overlay in order to make it intelligible, the instruction received from a host computer will be analyzed in fact, and these processings will be advanced according to a command. The procedure included to the command analysis is with the flow chart shown in drawing 4.

[0037] <Generation processing of page object> drawing 4 is the flow chart which showed the generation procedure of a page object including generation and registration of a form raster, and activation of a form overlay.

[0038] In this drawing, a control program 203 memorizes first the print data sent from a host computer 200 at step 401 to a receive buffer 207. Processing is ended, when it investigates whether print data exist (step S402) and does not exist in a receive buffer 207. On the other hand, when print data exist, print data are analyzed less than [step S403] continuing. First, at step S403, the print data under processing investigate whether it is a delivery instruction, and when it is judged that it is not a delivery instruction, the analysis of print data is continued less than [step S404] continuing. In addition, in step S403, although it explained that it investigated whether it is a delivery instruction (form feed command) since it was easy, in fact, it is an alphabetic character printing impaction efficiency instruction etc., and, in the printing data which can become the factor which causes delivery actuation which crosses the page lower limit under printing, progresses to henceforth [step S408] similarly.

[0039] step S404 — said print data — an overlay registration initiation instruction — or it investigates whether it is [overlay] under registration, and when it comes out so and is, in order to write form data in form data storage section 205b (step S406) and to read the continuing print data from a receive buffer 207, it returns to step S401, until it receives an overlay registration termination instruction (step S405). In addition, in order to judge whether it is [form data] under registration in step S404, the flag reset by 0 is prepared for the power up etc., when said flag is set to 1 and said flag is set at the time of overlay registration initiation instruction reception, it judges that it is [form data] under registration, and said flag is reset at the time of overlay registration termination instruction reception (S405). Or the size of the form data which should be registered as a parameter of an overlay registration initiation instruction is received as a counter,

and you may constitute so that form data may be registered, until said counter is set to 0. In addition, as the overlay storing section, nonvolatile memory, such as a flash memory, is prepared, and when it constitutes so that form data may be beforehand registered before print-data reception, each step of step S404 and step S406 may not be performed.

[0040] When it is judged in step S404 that it is not [form data] under registration When these print data are judged to investigate whether it is a form overlay run command, and to be a form overlay run command in continuing step S407 After setting to drawing object storing section 205a the information (after-mentioned) which shows that the form data which should be laid on top of the usual page under processing exist, It investigates whether the form raster corresponding to form data is registered in 205d of form raster storing sections (step S408). When having not registered, said form data are read from overlay storing section 205a at step S409, and drawing object generation section 203a generates the drawing object corresponding to each print data in form data (alphabetic data is included) (step S409).

[0041] In addition, although the analysis of the data usually printed by the page and the same processing as generation of a drawing object are sufficient as generation of the drawing object in step S409 (it mentions later at step S411), the drawing object of a page shall usually be stored in another field (for example, work-piece memory 205d) for drawing objects other than a font pattern, and a font pattern is registered into a font cache field. In addition, the alphabetic data registered into the font cache is usually used also at the time of processing of page data, and is hit into a cache. Moreover, in step S409, the management information for discriminating this form from generation of a drawing object to coincidence is also united, and it memorizes-like at the time of work-piece MEMORIHE. The form ID (specified by the host) it is decided for every form that will be a meaning, a paper size, the printing direction of a form, data generation resolution, etc. are contained in said management information here.

[0042] In continuing step S410, after drawing drawing objects other than a font and generating a form raster (drawing object drawing section 203c), a form image is agreement-sized (203f of raster agreement-sized sections), and said agreement-sized finishing form image is further stored in 205d of form raster storing sections (203d of form memory management sections). That is, after drawing temporarily said each drawing object corresponding to form data (except for a font pattern) generated in S409 to up to page memory 205c, it stores in 205d of form raster storing sections, and the drawing object finally generated in work-piece memory is cleared. what is generated and stored after the form raster generated at this time has divided 1 page into two or more bands in the gestalt of this operation — carrying out — in addition — and division height is constituted so that it may become the minimum height which page memory (band raster) 205c can take. Thus, even when band raster 205c can take the three-stage of 256 dot / 512 dot / 1024-dot height by loading memory with constituting, since a form raster is generated and stored in 256-dot height, it can use the same form raster under different conditions. In addition, what is necessary is just to perform the procedure which draws a drawing object to a form raster completely like the procedure which usually carries out lath reservoir MORIHE drawing of the drawing object of a page at the time of cypridium.

[0043] On the other hand, when print data are judged not to be a form overlay run command at step S407, it analyzes as print data which usually draw to a page, and a drawing object is generated (step S411). Namely, when print data are alphabetic character printing instructions The font information corresponding to the text specified with print data is read from a font ROM 204. Generate a corresponding dot pattern (= drawing object), and printing positional information, width of face, height, drawing logic, etc. with information required in case it draws to a page raster Said dot pattern is stored in drawing object storing section 205a (when it is the graphic data and image datas other than an alphabetic character, the pseudo code corresponding to each printing instruction is generated similarly). On the other hand, when it is judged in step S408 that the form raster corresponding to form data is registered in the form raster storing section, in order to analyze the continuing print data, it returns to step S401.

[0044] In step S403, when print data are delivery instructions (or instruction equivalent to a delivery instruction) The overlay data existence information carried out drawing object storing section 205a HESETTO is investigated at step S407 (step S412). When the form data which should be carried out a form overlay exist After form raster drawing section 203e reads and decrypts the

agreement-ized form raster [finishing / generation] for one band from 203d of form raster storing sections at said steps S407-S410, it draws to band raster 205c (step S413). The font (it generated at step S409) object contained in form at continuing step S414 is read from the drawing object storing section by one band, and band RASUTAHE drawing is carried out at the time of order. Finally the object of a page is usually drawn to a band raster, and the drawing processing for one band is ended (step S415). In order to process all band repeats (step S416) and the print data of the page which will continue if all band termination is carried out, the above-mentioned steps S412-S415 are returned to step S401, until drawing for 1 page is completed.

[0045] In addition, although the drawing step (step S415) of a page object was usually divided with the drawing step (step S414) of the font object in form and being explained here, if a series of drawing information is read one by one so that it may mention later by explanation of drawing 5, it is constituted so that each processing of steps S414 and S415 may be performed automatically.

[0046] Drawing 5 is the block diagram (drawing 5 (b), 5 (c)) showing the outline of the memory map (drawing 5 (a)) in which 1 page in drawing object storing section 205a of a page is usually shown and configuration in the gestalt of this operation. Setting to drawing 5 (a), a header unit 501 stores the various information on each page, and consists of existence information 502 grades of the information which shows the condition of the pages in the page number counted from a power up, printing resolution, a paper size, the number of band rasters, the total capacity of the drawing object belonging to this page, and drawn/cyripedium etc., and the overlay data which should draw. As for the band table 503, the drawing information 504 corresponding to each drawing object which only the number of said band rasters should be prepared and should be printed in each band is linked (drawing 5 (b)). Each drawing information and a drawing object are generated and secured at the drawing information storing section 504 and the drawing object storage section 505, respectively.

[0047] Drawing 5 (c) is the schematic diagram showing the configuration of one drawing information, and it is constituted so that the following information may be included as information for drawing a drawing object. Namely, the class of drawing object, the printing location in a band (bit offset value from an upper left edge), The drawing height of a drawing object, drawing logic with a band raster, the start address of a corresponding drawing object, The amount of offset which skips this drawing object when a drawing object is drawn from the band middle (the number of Rhine), The start address (the link last is taken as NULL) of the continuing drawing information etc. is contained (each information should have only the number of bits which can express an assignment form in assignment resolution at least). Here, in this invention, although the class of the above-mentioned drawing object shall consist of a binary bit map, a multiple-value image, a rectangle, graphic data encoded by the run length format, a compression font pattern, and Rhine, if expressed as an intermediate form which can generate the bit image for carrying out printer en JINHE sending out, it will not specify especially the format of a drawing object itself.

[0048] In step S415 of drawing 4, for every band, the processing which usually draws the drawing object of a page and the font object contained in form reads the drawing information linked from the above-mentioned band table one by one, and draws a drawing object according to the printing positional information within each drawing information etc. Since it is usually generated ahead of a page object, drawing information is also linked previously and the font object in form is previously drawn as a result. Moreover, in step S410 of drawing 4, when generating a form raster, it becomes generable [which is considering as the completely same configuration as the above, and used the same drawing object drawing section 203c]. [generable] [it] However, the drawing information corresponding to each font pattern is not linking, and a font pattern is not drawn in a form raster by the font pattern generate time in form data.

[0049] Drawing 6 is a memory block diagram which is managed by 203d of form memory management sections and in which showing the outline of the memory configuration of 205d of form raster storing sections. Setting to this drawing, the managed table 601 for every form consists of information required in order to draw and identify each form. The storage region for the managed table 601 may be beforehand secured in a fixed area, and it may be constituted so that it may be secured dynamically, as possible. In the managed table 601, the start address of the form ID for identifying each form (it manages by the same number as a host computer), printing resolution, a division band total, the sense of a form, a bandwidth, and the form raster band table

602 is contained. The form raster band table 602 is a table which was divided in step S410 of drawing 4 and which stores the information on a band-like form raster unit, and only the number according to the number of bands in the managed table 601 is secured, and it contains the start address of each form raster, and the height of each band. In addition, as long as height other than the band of the lowest edge is fixed, said constant value and the lowest edge band height may be held in the form managed table 601. Moreover, the sense and bandwidth of resolution and a form are set among the information stored in the managed table 601 based on print data or panel default value, and Form ID is directed as a parameter of an overlay registration instruction by the printer driver on a host etc.

[0050] Drawing 7 is the flow chart which showed the procedure of investigating whether the form raster corresponding to the form data with which 203d of form memory management sections was demanded being registered in the form raster storing section. That is, it is the procedure which showed the detail of step S408 of drawing 4. First, in step S701 of this drawing, it investigates whether it is in agreement with each of the form information which read the 1st form raster information and was specified by the form overlay run command from the form managed table 601 in 205d of form raster storing sections (steps S702–S706). That is, if not the same, in order to compare Form ID (step S702), to compare the following form information if the same, and to read the following form raster information, it progresses to step S708. Like the following the paper-size case by which the form conveyance direction of printing resolution (step S703), and a lengthwise direction/longitudinal direction is supported The sense of a form (step S704), In the case of a color printer, (step S706) is investigated for whether it is ***** without color mode (step S705) and page expansion / contraction assignment, respectively. When all the conditions are in agreement, a form number is memorized as form raster existence information at step S707, and it is judged that the form raster corresponding to an assignment form overlay is registered.

[0051] A form raster is judged [having not registered and] when retrieval of return and all form rasters is completed to step S701 on the other hand, if it investigates whether retrieval of all form rasters was completed and has not ended at step S708, when one information of steps S702–S706 is not in agreement (step S709).

<Generation of form raster from ROM data> drawing 8 is a flow chart which shows a form image generation procedure in case the form data of a Page Description Language format are storing ending in the non-volatile device (this device is hereafter explained as a flash ROM) of a flash ROM, a hard disk, etc. which can be written in. That is, when form data are offered by the flash ROM, a form raster is generated and held not only by the procedure of drawing 4 R> 4 but by the procedure of drawing 8.

[0052] Before performing at the idle time which has not received the print data which should be processed before receiving print data or usually receiving the print data of a page immediately after supplying a power source to an image output unit, processing of this flow chart may be constituted so that a form raster generation instruction may be received and executed from a host computer.

[0053] In this drawing, it investigates first whether it is equipped with the flash ROM at step S801, and when not mounted, processing is ended. On the other hand, when mounted, the printing conditions (printing resolution, the form conveyance direction, color mode, etc.) set up with the panel menu etc. at step S802 are read, and henceforth [continuing step S803], a form raster is generated from the form data memorized by the flash ROM, and it registers to the form raster storing section. First, it investigates whether the form data in a flash ROM are already registered into the form raster storing section (step S803), if it has not registered, form data will be read (step S804), the drawing object for 1 page containing a font pattern is generated (step S805), and each font pattern is registered to a font cache means.

[0054] When it investigates whether print data were received and print data exist in a receive buffer at continuing step S806, the drawing object [finishing / generation] corresponding to form data is deleted (step S814), and after releasing the work-piece memory which was being used as activity memory, form image generation processing is ended. Here, if it is in the middle of storing of a form image, the form image stored to the middle will also be released. In addition, with the gestalt of this operation, only in step S806, although existence of received data was checked, the step [not only this but] order which the processing time cuts in many, and the thing which do not use

form by checking print data periodically and carrying out like this with a timer etc. and for which printing processing of data is usually barred are lost comparatively.

[0055] After drawing the form image for one band to a band buffer at continuing step S807 using said drawing objects other than a font object, said form image is agreement-ized at step S808. In continuing step S809, when it investigates whether memory space required for registration is in the form image storing section and **** memory space runs short, processing is ended, after performing an error message at step S810. On the other hand, if registration is possible, an agreement-ized finishing form image is stored (step S811). The drawing object which became unnecessary [other than a font cache] at step S813 at the last is deleted repeatedly (step S812) until it all ends processing of the above-mentioned steps S806-S811 by 1 page, and processing of 1 form is ended. In addition, with the gestalt of this operation, although deed processing was interrupted for the error message at the time of out of memory, the priority of each form may be specified by panel setup etc., or it may constitute so that operating frequency may be memorized with form data, and it may delete from a form image storing field in the low order of priority, or order with low operating frequency, and processing of this form image storing may be continued.

[0056] With the above procedure, the form data which were registered by form data registration and which were described by the Page Description Language or the form data which is beforehand offered by ROM and which was described by the Page Description Language is changed into the drawing object by middle data with a form data generation instruction, and the raster data of the form are generated further, it encodes, and it is memorized. If it becomes clear that overlay is during delivery of the page which has received the delivery instruction at the page, the encoded form raster data which are memorized will be decoded, it will develop in band memory, and usual page objects other than form will be developed and printed on it.

[0057] Moreover, since it is held in the format developed by bit map data together with form data in a font cache about the alphabetic data which constitutes the form, about form data, the cache is carried out also including the alphabetic character contained in it, and it becomes possible to reuse.

[0058] By doing in this way, since the once registered form data are memorized as encoded raster data, it is not necessary to develop from a Page Description Language, and they can be quickly developed in band memory.

[0059] In addition, what is necessary is to only exclude the expansion processing to the raster data, to encode as it is and just to store it, when form data are transmitted not by the Page Description Language but by the image data.

[0060] With the gestalt of [gestalt of the 2nd operation] the 1st operation, since it registered with the font cache means fair [the alphabetic data contained in form data, and the alphabetic data usually used on a page], the font cache used within form may have been deleted by insufficient memory etc.

[0061] Therefore, it constitutes so that it may be hard to be deleted compared with the alphabetic data with which the font cache currently used within form data by setting up more highly [priority] than the alphabetic data usually contained in a page to the font pattern generated and registered from the alphabetic data contained in the form data among the font patterns registered into the font cache means with the gestalt of this operation is usually used only within the page.

[0062] Drawing 9 is a block diagram explaining the basic configuration of the image output unit in which the gestalt of operation of the 2nd of this invention is shown, gives the same number about the same configuration as the gestalt of the 1st operation, and omits explanation. In this drawing, in case priority setup / judgment section 901 registers 203g HEFONTO pattern of font cache Management Department, it sets up priority, and it manages from which font pattern it deletes at the time of font pattern deletion. The configuration of those other than this is the same as that of drawing 2.

<Structure of font cache where priority was attached> drawing 10 R> 0 is the schematic diagram showing the configuration of the font cache stored in the font cache storing section in the gestalt of this operation.

[0063] In the gestalt of this operation, by using a character code as a search key, each font pattern 1006 follows indexes 1002 and 1003, and can search them now from the font information management table 501 (henceforth, managed table). Here, the character code given from the

outside is 1 byte or 2 bytes, and an index has two steps of depth, the object (1002) for the high-order bytes of a character code, and the object (1003) for lower bytes. When registering a font pattern to a font cache, the pointer to the low order index or font cache stereo following the location corresponding to each of the high-order byte/lower byte of a character code is written in (initial value in an index is set to NULL). On the other hand, at the time of font cache retrieval, first, it investigates whether the pointer of the index for lower bytes is set to the location of the index corresponding to the high-order byte of said character code, and when set, a low order index is searched with the lower byte of a character code, and the pointer to the stereo of a font cache is obtained similarly. In addition, the index for high-order bytes is constituted only from the member of code =0x00 by the coding scheme of a single cutting tool system.

[0064] In this drawing, the following information is stored in the managed table 1001 for every font as information for identifying each font. Namely, a typeface name, the graphic set in which allotment of a character code and a real pattern is shown, The flag which shows whether they are a character size and a scalable font, the stroke wait which shows the size of the line of an alphabetic character (un-illustrating), Others [flag / (un-illustrating) / the flag (un-illustrating) which shows a fixed pitch or a proportional pitch, and / which shows columnar writing/lateral writing], The pointer to the high order index 1002 for searching a font pattern etc. is stored, and the font number is assigned for every managed table (in drawing 10 , they are No. 1 and No. 2). In this drawing, the case where the double cutting tool system font is assigned and the font pattern corresponding to character code =0x2422 and 0x3021 is linked is shown in the font number =1 No. managed table.

[0065] The font cache 1004 consists of the header information section 1005 and a font pattern 1006. In addition to information required for drawing of the dot width of face w of a font pattern, dot height h, the start address of angle of rotation and a font pattern and the offset value from the base line, pitch information, etc., the font information used as the retrieval judging key of font caches, such as expansion and qualification information of the existence of board qualification, is set to the header information section 1005. Moreover, when the character pattern is compressed and stored, in addition to the above-mentioned information, it also has the agreement-sized flag which shows that it is compressed.

[0066] In the font cache which furthermore starts this invention, by having priority flag 1005a which shows whether this font cache was used in the fixed form form department, if it is during form data analysis, namely, if it is alphabetic data contained in form data, priority flag 1005a will be set 1 at font cache creation time, otherwise, 0 will be set (if it is the font cache usually created during page analysis). Moreover, when the font pattern which searches and asks for a font cache is found (it hit into the cache), it is during form data analysis, and when priority flag 1005a is set to 0, this priority flag is reset to 2. In addition, when priority flag 1005a is set to 2, it is shown that it is the font cache usually used from both pages in form.

[0067] On the other hand, when it hits to a font cache, it is usually during page data analysis, and also when the priority flag is set to 1, this priority flag is reset to 2, and it is shown that it is said font cache usually similarly used from both pages in form.

[0068] In addition, although drawing 10 explained the font pattern 1006 as a character pattern generated in RAM205, when it has a font pattern beforehand in a font ROM 204 like a dot font, a font pattern may not exist in RAM205. In this case, as for the start address of said font pattern in the header information section 1005, the address of the font pattern head in a font ROM 204 will be set.

[0069] In the font cache of a configuration of having explained above, the font cache which the font cache usually used only on the page is deleted by the 1st, and is continuously used only within form because a priority flag deletes in order of 0->1->2, and the font cache usually used for the last on both pages in form will be deleted at the times of font cache deletion, such as at the time of out of memory etc. When the font cache used in the fixed form form section becomes is hard to be deleted and it uses form especially by two or more pages with such a procedure, it becomes possible to raise the hit ratio of a font cache.

[0070] Drawing 11 is a flow chart which shows the expansion procedure of the font mentioned above. the memory space for developing a font — ** — in not being enough, it sets (step S111-YES) and a deletion threshold as 0 (step S112). Next, only the amount from which required

memory space can secure the font cache which has the priority below the deletion threshold or all any, or little way is deleted (step S113). If it judges whether required memory was securable by that cause (step S114) and can secure, a font will be developed using the memory (step S115). Moreover, it judges (step S116), it will still be said that memory space is insufficient whether if memory is not enough, a current deletion threshold is 2, even if it will delete all font caches, if it is 2, and this considers it as an error. Moreover, if a deletion threshold is 0 or 1, one current deletion threshold will be raised and the font cache of the following priority will be deleted.

[0071] Thus, when becoming that it is hard to be deleted and using form especially by two or more pages, the font cache used in the form section has the hit ratio of a font cache raised, and can raise the effectiveness of printing processing.

[0072] In addition, in order to show that priority 1 or the font cache of 2 is included for every class of font, the flag which indicates that the alphabetic character of a high priority is contained in the scalable flag column etc. may be formed. In this case, when a priority flag is set to 1 or 2, the flag which shows that the alphabetic character of this high priority (priority 1 or 2) is contained in coincidence is also set. Moreover, in deleting a font cache, first, for every class of font, whether the alphabetic character of a high priority is contained in the font judges, and if not contained, it deletes from the font. If it turns out that it is contained, other fonts will be investigated and the same processing will be repeated.

[0073] In addition, it does not matter even if it sets up the priority which deletes a font cache also using information, such as existence of qualification, font pattern size, and the 2nd level / level 1 kanji characters / the 1st level non-kanji character of JIS, in addition to the above-mentioned priority.

[0074]

[Other operation gestalten] In addition, even if it applies this invention to the system which consists of two or more devices (for example, a host computer, an interface device, a reader, a printer, etc.), it may be applied to the equipments (for example, a copying machine, facsimile apparatus, etc.) which consist of one device.

[0075] Moreover, the purpose of this invention supplies the storage which recorded the program code of the software which realizes the function of the operation gestalt mentioned above to a system or equipment, and is attained also by carrying out read-out activation of the program code with which the computer (or CPU and MPU) of the system or equipment was stored in the storage.

[0076] In this case, the function of the operation gestalt which the program code itself read from the storage mentioned above will be realized, and the storage which memorized that program code will constitute this invention.

[0077] As a storage for supplying a program code, a floppy disk, a hard disk, an optical disk, a magneto-optic disk, CD-ROM, CD-R, a magnetic tape, the memory card of a non-volatile, ROM, etc. can be used, for example.

[0078] Moreover, by performing the program code which the computer read, a part or all of processing that OS (operating system) which the function of the operation gestalt mentioned above is not only realized, but is working on a computer based on directions of the program code is actual is performed, and also when the function of the operation gestalt mentioned above by the processing is realized, it is contained.

[0079] Furthermore, after the program code read from the storage is written in the memory with which the functional expansion unit connected to the functional add-in board inserted in the computer or the computer is equipped, a part or all of processing that CPU with which the functional add-in board and functional expansion unit are equipped is actual performs, and also when the function of the operation gestalt mentioned above by the processing is realized, it is contained based on directions of the program code.

[0080]

[Effect of the Invention] As explained above, according to this invention, it holds as image data which encoded the form data for 1 page, and it becomes possible not to be concerned with the contents of form data, but to end drawing processing at a high speed by decrypting said form image corresponding to the time of form overlay run command reception. especially — the form data of the same contents — receiving — usually — a page — a ** page — ** — in the printing

instruction which is two or more pages [like], the time amount which generation of the form data after the 2nd page takes can be suppressed small.

[0081] Since it constituted so that the form image which does not contain alphabetic data especially might be agreement-ized, while raising the rate of agreement-izing of an agreement-ized means (compressibility) and raising the utilization ratio of memory, the processing time which agreement-izing and compound-ization take can be shortened.

[0082] Moreover, since it becomes unnecessary to have the same drawing object for each individual for every page, it becomes possible to stop low the fall of processing speed, such as waiting for delivery, and the memory which drawing takes.

[0083] Moreover, when form data are memorized by the non-volatile device, the page [1st] first print can also be outputted for the form image corresponding to form data to a high speed by generation and agreement-izing using the idle time which has not received the data which should be printed from a power up or a host.

[0084] moreover, it become applicable simply by hold about said form image in the unit which divided 1 page in the shape of a band to agreement[generation /]-izing and the image output unit which pretreatment and processing of a skip become unnecessary about said form image in the case of a decryption with the sequential decryption and having constitute so that it might begin to have read and might draw to a band buffer at the time of form overlay assignment, and usually draw page data by the banding approach.

[0085] Furthermore, by carrying out priority to the font pattern contained in form data among the font patterns registered into the font cache that it sets up highly and is hard to be deleted rather than the alphabetic data contained in usual pages other than form, it becomes easy to hit the font cache included in form data, and the processing speed in two or more pages form printing can be improved.

[0086]

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention relates to printers, such as the image output unit and approach in which an overlay output of page data and form data is for example usually possible, especially a laser beam printer.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] Atypical data divisions in which the contents differ from the fixed form form sections (a title, closing line, etc.) which print the same contents over two or more pages for every page like a document or a fixed form document in printing on general business (a figure, identifier, etc.) usually, page data — calling — form overlay printing piled up and printed is used in many cases.

[0003] Then, in the conventional page printer which analyzes a Page Description Language and generates a raster image, it registered with the storage region of a printer by making into event drive macro instruction the fixed form form section created in said Page Description Language format, and it is carrying out automatic activation of said macro by events, such as delivery actuation, and form overlay printing was realized. The time of assignment of this macro (1st activation), and after the analysis of the usual page data for 1 page is completed, whenever it receives a delivery instruction (2nd henceforth), the contents of registration are read, and said event drive macro is usually analyzed like page data by the Page Description Language analysis means, and is changed into the pseudo code (it is hereafter called a drawing object) for generating the raster image for 1 page.

[0004] Moreover, it may be constituted so that said form data (macro data) may be beforehand transmitted to a printer from a host computer and it is not necessary to transmit form data by registering to nonvolatile memory, such as a flash memory, each time in advance of printing activation. In this case, only the instruction which directs the form data in a flash memory as an event drive macro will be sent at the time of printing activation.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] As explained above, according to this invention, it holds as image data which encoded the form data for 1 page, and it becomes possible not to be concerned with the contents of form data, but to end drawing processing at a high speed by decrypting said form image corresponding to the time of form overlay run command reception. especially — the form data of the same contents — receiving — usually — a page — a ** page — ** — in the printing instruction which is two or more pages [like], the time amount which generation of the form data after the 2nd page takes can be suppressed small.

[0081] Since it constituted so that the form image which does not contain alphabetic data especially might be agreement-ized, while raising the rate of agreement-izing of an agreement-ized means (compressibility) and raising the utilization ratio of memory, the processing time which agreement-izing and compound-ization take can be shortened.

[0082] Moreover, since it becomes unnecessary to have the same drawing object for each individual for every page, it becomes possible to stop low the fall of processing speed, such as waiting for delivery, and the memory which drawing takes.

[0083] Moreover, when form data are memorized by the non-volatile device, the page [1st] first print can also be outputted for the form image corresponding to form data to a high speed by generation and agreement-izing using the idle time which has not received the data which should be printed from a power up or a host.

[0084] moreover, it become applicable simply by hold about said form image in the unit which divided 1 page in the shape of a band to agreement[generation /]-izing and the image output unit which pretreatment and processing of a skip become unnecessary about said form image in the case of a decryption with the sequential decryption and having constitute so that it might begin to have read and might draw to a band buffer at the time of form overlay assignment, and usually draw page data by the banding approach.

[0085] Furthermore, by carrying out priority to the font pattern contained in form data among the font patterns registered into the font cache that it sets up highly and is hard to be deleted rather than the alphabetic data contained in usual pages other than form, it becomes easy to hit the font cache included in form data, and the processing speed in two or more pages form printing can be improved.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

TECHNICAL PROBLEM

[The technical problem which invention will solve and to carry out] However, there was a trouble that it explained below, in the above-mentioned conventional example.

[0006] That is, since form data were registered in a Page Description Language format when said event drive macro has realized form overlay printing, in the case of complicated form data, the analysis of the form data itself and generation of a drawing object had taken time amount. Namely, the more only processing speed equivalent to the case where the output of the 1st page usually outputs the contents same as data, without using overlay was obtained and the contents of form became complicated, the more the processing time of the form section was increasing.

[0007] Moreover, in case the print data which are using the same form by two or more pages are outputted, in order to analyze the same macro (command group) for every page each time and to generate the same drawing object, in spite of having been the page of the same contents, there was a trouble that the printing speed after the 2nd page did not improve. That is, since ** page generation was carried out, sharing of a drawing object was not able to do the same drawing object in two or more pages. Therefore, memory required for two or more pages drawing processing increased, and we were anxious also about the performance fall of the waiting for delivery occurring. When output specification of the overlay is carried out, the error of drawing insufficient memory etc. also becomes easy to generate the complicated form which contains many graphic forms and image images especially, and solution of the above-mentioned technical problem was important.

[0008] As one of the solutions of the above-mentioned problem, form data are rasterized beforehand, it holds in the printer as an image data (form image), and it is possible to usually pile up with page data and to draw. However, in order to hold the image for 1 page, a lot of memory is needed, and it is not practical (it is A4, 600dpi, and monochrome and about 8MB of memory is the need).

[0009] Then, although it was possible to agreement-ize a form image (compression), since the compressibility of the pattern containing an alphabetic character generally was not good, the pack BITTSU compression (an image is agreement-ized paying attention to the continuity of a byte order) which can be adapted for compression of a form image had the problem that the compressibility of a form image will fall, when many alphabetic data was contained in form data.

[0010] Even if it was made in view of the above-mentioned trouble of this invention and the contents of form become complicated, the processing time of the form section is not increased, and if it is the form of the same contents, the printing speed after the 2nd page can be raised, and memory required for the drawing processing which is two or more pages is not increased, and it aims at offering the ***** output unit and approach of not reducing performance, such as waiting for delivery.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] This invention for solving the above-mentioned trouble is equipped with the following configurations. Namely, a form storing means to encode and store the form data generated as bit map data which do not contain an alphabetic character, A form generation means to generate form data as bit map data, to encode, and to store in said form storing means, The font cache which stores the alphabetic data generated as bit map data, An alphabetic character generation means to generate alphabetic data as bit map data, and to store in said font cache, In case print data are overlaid with form data and printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is an image output unit equipped with an overlay means to draw using the bit map data stored in said font cache, and to output said print data in piles.

[0012] Or the form generation process which generates the form data which do not contain an alphabetic character as bit map data, encodes, and is stored in a form storing means, The alphabetic character generation process which generates alphabetic data as bit map data, and is stored in a font cache, in case it overlays print data with form data and they are printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is an image output method equipped with the overlay process which draws using the bit map data stored in said font cache, and outputs said print data in piles.

[0013] Or a form generation means to generate the form data which do not contain an alphabetic character as raster data, to encode, and to store in a form storing means, An alphabetic character generation means to generate alphabetic data as bit map data, and to store in a font cache, in case it overlays print data with form data and they are printed, while decoding the stored form data which were encoded The alphabetic data contained in form data is the computer-readable memory which stored the program operated as a computer as equipment which has an overlay means to draw using the bit map data stored in said font cache, and to output said print data in piles.

[0014]

[Embodiment of the Invention] Before explaining the configuration of this operation gestalt, the configuration of a suitable image processing system to apply this invention is explained referring to drawing 1 . Drawing 1 is the sectional view showing the configuration of the image processing system which can apply this invention, for example, shows the case of a laser beam printer (henceforth, LBP). In addition, the image processing system which can apply this invention is not restricted to LBP, and it cannot be overemphasized that you may be printer equipment of other print methods.

[0015] in this drawing , the LBP body 100 create a character pattern , a graphic form pattern , etc. which correspond according to those information , and form an image in the record paper which be a record medium while it input and memorize the text and graphic forms (character code etc.) information which be supply in the form of a Page Description Language , fixed form format data (form data) or other macro instruction , etc. from the host computer (200 of drawing 2) connect outside . As for the control panel 112, a switch, an LED drop, etc. for actuation are allotted. The printer control unit 101 analyzes the text supplied from control and the host computer of the LBP100 whole. Text is changed into the corresponding video signal of a character pattern, and this control unit 101 mainly outputs it to a laser driver 102. Moreover, this LBP can also perform

registration of registration of a character pattern, form data, etc. from the non-illustrated source of data, is similarly changed into a video signal according to printing data, and is outputted to a laser driver 102. A laser driver 102 is a circuit for driving semiconductor laser 103, and carries out the on-off change of the laser beam 104 discharged from semiconductor laser 103 according to the inputted video signal. Laser 104 is shaken at a longitudinal direction by the rotating polygon 105, and scans the electrostatic drum 106 top. Thereby, the electrostatic latent image of a character pattern is formed on the electrostatic drum 106. After this latent image is developed by the development unit 107 of electrostatic drum 106 perimeter, it is transmitted to the recording paper. - uses a cut sheet for this recording paper, and the cut sheet recording paper is contained by the form cassette 108 with which LBP100 was equipped, is incorporated in equipment with the feed roller 109 and the conveyance rollers 110 and 111, and is supplied to the electrostatic drum 106.

[0016] With reference to an attached drawing, the gestalt of operation of the 1st of this invention is explained below to [the gestalt of the 1st operation].

[0017] Drawing 2 is a block diagram explaining the basic configuration (internal configuration of the control unit 101 of the image processing system of drawing 1) of the image output unit in which the gestalt of operation of the 1st of this invention is shown, for example, shows the case of a laser beam printer.

[0018] Moreover, as long as the function of this invention is performed, even if it is the device of a simple substance, you may be the system by which processing is performed through networks, such as LAN.

[0019] In this drawing, it connects with the external host computer 200 through the predetermined interface 208, and the body 100 of a printer is constituted so that predetermined printing processing may be performed according to the print data sent from a host computer 200. These print data received from the host computer 200 are temporarily stored in a receive buffer 207 as received data.

[0020] In addition, the form data described in the Page Description Language format are stored in the hard disk of a host computer 200, and each form data are matched with the information on the identifiable form ID etc., and are managed by the meaning. Moreover, said form data are sent out as a form registration instruction (event drive macro) at the time of form overlay printing.

[0021] A printer CPU 201 controls in generalization access with the various devices connected to the system bus 206 based on the control program 203 memorized by ROM202 which is a read-only memory. Analyze the print data stored in the receive buffer 207, and the drawing object for at least 1 page is generated (drawing object generation section 203a). Then, the raster image for one band divided into 1 page or two or more bands based on said drawing object is generated (drawing object drawing section 203c). The video signal by ON/OFF of the bit of said raster image is outputted to the printing section 210 (printer engine) through printing section I/F and 209 (cypripedium is carried out). Performing the printing section 210 based on the video signal received from CPU201, it prints an image on the recording paper 211. Here, the font scaler which generates the font pattern corresponding to alphabetic data (a typeface, size, character code) is also contained in drawing object generation section 203a.

[0022] In addition, when preparing a part for at least 2 bands which divided the page in the shape of a band, without preparing the memory which stores a raster image by 1 page (henceforth, band raster), it is constituted so that generation and cypripedium actuation of said raster image may be performed to coincidence. That is, the drawing object belonging to the band raster of another side is drawn, carrying out the cypripedium of one band raster [finishing / drawing] (the above-mentioned drawing procedure is hereafter called drawing by banding).

[0023] Moreover, the control program 203 consists of programs for realizing the following functions in addition to the above-mentioned configuration.

[0024] Namely, when the print data in a receive buffer 207 are form registration instructions The overlay registration / read-out section 203b which registers said form data described by the Page Description Language into RAM205 (after-mentioned), and reads said form data at the time of form overlay run command reception, Drawing object drawing section 203c to which the form raster which does not contain an alphabetic character based on the read form data is generated, and alphabetic data generates a font pattern, 203d of form memory management sections which

memorize and hold said form raster in RAM205, 203g of font cache Management Department which reuses a registered font pattern, without using a font scaler when said font pattern is registered to the font cache storing section and the output of the same alphabetic data is required, 203f of form compression / expanding sections decrypted before agreement-izing before memorizing said form raster, usually piling up with a page and drawing. It consists of form raster drawing section 203e which usually piles up a form raster [finishing / said decryption] with page data, and draws to page memory 205c.

[0025] In addition, the form raster corresponding to the demanded form data investigates whether it is finishing [storage in 205d of form raster storing sections], and 203d of said form memory management sections is equipped with the cache function to use a memorized form raster when it is already storage ending (however, in order to use a font cache [finishing / generation] about the alphabetic data in form data, read-out of form data is performed from overlay storing section 203a). Moreover, this ROM202 is used also as memory which stores the font data which consists of a dot font used for an alphabetic character output, and a scalable font (font ROM 204).

[0026] RAM205 is RAM which functions as the main memory of CPU201, work-piece memory, etc., and it is constituted so that memory space can be extended with the option RAM connected to the extension port which is not illustrated. Drawing object storing section 205a (the font cache registration memory which carries out the cache of the character pattern developed based on the scalable data in said font memory 204 is included) in which RAM205 stores said drawing object, Work-piece memory 205e temporarily used by control program 203a, Page memory 205c which stores the raster image (or raster image for 1 page) equivalent to two regions of the band field which divided 1 page in the shape of a band. It is used for printing environmental data storage memory besides 205d of form raster storing sections for agreement-izing the form image which analyzed form data storage section 205b which stores the form data described by the Page Description Language, and said form data, and was obtained, and holding it etc. (un-illustrating). In addition, LBP100 has received supply of power from the power supply section which does not illustrate. Moreover, although it explained that form data were stored in RAM205 above, you may store in nonvolatile memory, such as a flash memory, beforehand.

[0027] In addition, although it explained that the equipment which constitutes this invention was supplied as a program stored in ROM, after being supplied by media, such as not only this but a floppy disk, and a hard disk, and being loaded to RAM etc. before activation, it may be constituted so that it may perform.

The procedure of the overlay output in a <overlay output-processing procedure>, then the image output unit by the above-mentioned configuration is explained below using the flow chart of drawing 3 and drawing 4.

[0028] When <outline of form data registration> drawing 3 (A) receives form data from a host computer 200, it is the flow chart which showed the outline of the instruction procedure for generating and registering a form raster.

[0029] First, in step S301, if a form registration instruction is received, form data will be registered to form data storage section 205b in a printer. Here, a form registration instruction means a series of instruction groups which consist of live data of an overlay registration initiation instruction - form - an overlay registration termination instruction. Then, according to a form lath evening generation instruction, the form data registered at step S301 are read (step S302), and the drawing object which analyzes form data and contains the corresponding font pattern for 1 page is generated (step S303). In addition, the generated font pattern is usually registered to the font cache like the alphabetic data of a page.

[0030] Then, drawing objects other than a font object are drawn to a band raster, and a form raster is generated (step S304). If generation of a form raster is completed, after agreement-izing said form raster at step S305, a form raster is registered with the information on the form ID for identifying form etc., and form raster registration processing is ended.

[0031] <Outline of form overlay activation> drawing 3 (B) is the flow chart which showed the outline of the procedure at the time of receiving a form overlay run command. In addition, reception and processing of registration of the form data shown in drawing 3 (A) and form overlay activation of drawing 3 (B) may be done continuously (with the same print job), and it does not matter even if it performs only registration of form data previously.

[0032] In this drawing, it investigates first whether the form raster corresponding to the form ID specified by the form overlay run command is registered in step S310 (203d of form raster Management Department). When already registered, the directions which draw this form raster are told to form raster drawing section 203e (step S312), only a part for the alphabetic character data division of form data is analyzed to coincidence, and a font object is generated (step S313). Since a font object hits to the font cache registered at step S303 of drawing 3 (A) at this time, it is not carried out by the scaling processing by the scaler.

[0033] On the other hand, when judged [having not registered a desired form raster at step S310, and], at step S311, all form data are read and a drawing object is generated. What is necessary is just to perform the generation of a drawing object itself like the time of form raster registration.

[0034] After processing of form data is completed as mentioned above, the usual page data of the atypical section are analyzed and a drawing object is generated like step S311 etc. (S314).

[0035] After usually processing page data with the form data for 1 page, band RASUTAHE drawing of the form part is carried out (step S315). That is, the font object contained in an agreement-sized form raster and form when a form raster is registered is drawn, and on the other hand, when having not registered, the drawing object group (a font object is included) equivalent to a form part is drawn to a band raster. Finally, the drawing object of a page is usually drawn similarly (step S316), and the executive operation of a form overlay is ended.

[0036] Now, although the flow chart shown in drawing 3 shows independently the procedure which shows registration of form, and the procedure which shows activation of a form overlay in order to make it intelligible, the instruction received from a host computer will be analyzed in fact, and these processings will be advanced according to a command. The procedure included to the command analysis is with the flow chart shown in drawing 4.

[0037] <Generation processing of page object> drawing 4 is the flow chart which showed the generation procedure of a page object including generation and registration of a form raster, and activation of a form overlay.

[0038] In this drawing, a control program 203 memorizes first the print data sent from a host computer 200 at step 401 to a receive buffer 207. Processing is ended, when it investigates whether print data exist (step S402) and does not exist in a receive buffer 207. On the other hand, when print data exist, print data are analyzed less than [step S403] continuing. First, at step S403, the print data under processing investigate whether it is a delivery instruction, and when it is judged that it is not a delivery instruction, the analysis of print data is continued less than [step S404] continuing. In addition, in step S403, although it explained that it investigated whether it is a delivery instruction (form feed command) since it was easy, in fact, it is an alphabetic character printing impaction efficiency instruction etc., and, in the printing data which can become the factor which causes delivery actuation which crosses the page lower limit under printing, progresses to henceforth [step S408] similarly.

[0039] step S404 — said print data — an overlay registration initiation instruction — or it investigates whether it is [overlay] under registration, and when it comes out so and is, in order to write form data in form data storage section 205b (step S406) and to read the continuing print data from a receive buffer 207, it returns to step S401, until it receives an overlay registration termination instruction (step S405). In addition, in order to judge whether it is [form data] under registration in step S404, the flag reset by 0 is prepared for the power up etc., when said flag is set to 1 and said flag is set at the time of overlay registration initiation instruction reception, it judges that it is [form data] under registration, and said flag is reset at the time of overlay registration termination instruction reception (S405). Or the size of the form data which should be registered as a parameter of an overlay registration initiation instruction is received as a counter, and you may constitute so that form data may be registered, until said counter is set to 0. In addition, as the overlay storing section, nonvolatile memory, such as a flash memory, is prepared, and when it constitutes so that form data may be beforehand registered before print-data reception, each step of step S404 and step S406 may not be performed.

[0040] When it is judged in step S404 that it is not [form data] under registration When these print data are judged to investigate whether it is a form overlay run command, and to be a form overlay run command in continuing step S407 After setting to drawing object storing section 205a the information (after-mentioned) which shows that the form data which should be laid on top of

the usual page under processing exist, It investigates whether the form raster corresponding to form data is registered in 205d of form raster storing sections (step S408). When having not registered, said form data are read from overlay storing section 205a at step S409, and drawing object generation section 203a generates the drawing object corresponding to each print data in form data (alphabetic data is included) (step S409).

[0041] In addition, although the analysis of the data usually printed by the page and the same processing as generation of a drawing object are sufficient as generation of the drawing object in step S409 (it mentions later at step S411), the drawing object of a page shall usually be stored in another field (for example, work-piece memory 205d) for drawing objects other than a font pattern, and a font pattern is registered into a font cache field. In addition, the alphabetic data registered into the font cache is usually used also at the time of processing of page data, and is hit into a cache. Moreover, in step S409, the management information for discriminating this form from generation of a drawing object to coincidence is also united, and it memorizes-like at the time of work-piece MEMORIHE. The form ID (specified by the host) it is decided for every form that will be a meaning, a paper size, the printing direction of a form, data generation resolution, etc. are contained in said management information here.

[0042] In continuing step S410, after drawing drawing objects other than a font and generating a form raster (drawing object drawing section 203c), a form image is agreement-ized (203f of raster agreement-ized sections), and said agreement-ized finishing form image is further stored in 205d of form raster storing sections (203d of form memory management sections). That is, after drawing temporarily said each drawing object corresponding to form data (except for a font pattern) generated in S409 to up to page memory 205c, it stores in 205d of form raster storing sections, and the drawing object finally generated in work-piece memory is cleared. what is generated and stored after the form raster generated at this time has divided 1 page into two or more bands in the gestalt of this operation — carrying out — in addition — and division height is constituted so that it may become the minimum height which page memory (band raster) 205c can take. Thus, even when band raster 205c can take the three-stage of 256 dot / 512 dot / 1024-dot height by loading memory with constituting, since a form raster is generated and stored in 256-dot height, it can use the same form raster under different conditions. In addition, what is necessary is just to perform the procedure which draws a drawing object to a form raster completely like the procedure which usually carries out lath reservoir MORIHE drawing of the drawing object of a page at the time of cyripedium.

[0043] On the other hand, when print data are judged not to be a form overlay run command at step S407, it analyzes as print data which usually draw to a page, and a drawing object is generated (step S411). Namely, when print data are alphabetic character printing instructions The font information corresponding to the text specified with print data is read from a font ROM 204. Generate a corresponding dot pattern (= drawing object), and printing positional information, width of face, height, drawing logic, etc. with information required in case it draws to a page raster Said dot pattern is stored in drawing object storing section 205a (when it is the graphic data and image datas other than an alphabetic character, the pseudo code corresponding to each printing instruction is generated similarly). On the other hand, when it is judged in step S408 that the form raster corresponding to form data is registered in the form raster storing section, in order to analyze the continuing print data, it returns to step S401.

[0044] In step S403, when print data are delivery instructions (or instruction equivalent to a delivery instruction) The overlay data existence information carried out drawing object storing section 205a HESETTO is investigated at step S407 (step S412). When the form data which should be carried out a form overlay exist After form raster drawing section 203e reads and decrypts the agreement-ized form raster [finishing / generation] for one band from 203d of form raster storing sections at said steps S407-S410, it draws to band raster 205c (step S413). The font (it generated at step S409) object contained in form at continuing step S414 is read from the drawing object storing section by one band, and band RASUTAHE drawing is carried out at the time of order. Finally the object of a page is usually drawn to a band raster, and the drawing processing for one band is ended (step S415). In order to process all band repeats (step S416) and the print data of the page which will continue if all band termination is carried out, the above-mentioned steps S412-S415 are returned to step S401, until drawing for 1 page is completed.

[0045] In addition, although the drawing step (step S415) of a page object was usually divided with the drawing step (step S414) of the font object in form and being explained here, if a series of drawing information is read one by one so that it may mention later by explanation of drawing 5, it is constituted so that each processing of steps S414 and S415 may be performed automatically.

[0046] Drawing 5 is the block diagram (drawing 5 (b), 5 (c)) showing the outline of the memory map (drawing 5 (a)) in which 1 page in drawing object storing section 205a of a page is usually shown and configuration in the gestalt of this operation. Setting to drawing 5 (a), a header unit 501 stores the various information on each page, and consists of existence information 502 grades of the information which shows the condition of the pages in the page number counted from a power up, printing resolution, a paper size, the number of band rasters, the total capacity of the drawing object belonging to this page, and drawn/cyripedium etc., and the overlay data which should draw. As for the band table 503, the drawing information 504 corresponding to each drawing object which only the number of said band rasters should be prepared and should be printed in each band is linked (drawing 5 (b)). Each drawing information and a drawing object are generated and secured at the drawing information storing section 504 and the drawing object storage section 505, respectively.

[0047] Drawing 5 (c) is the schematic diagram showing the configuration of one drawing information, and it is constituted so that the following information may be included as information for drawing a drawing object. Namely, the class of drawing object, the printing location in a band (bit offset value from an upper left edge), The drawing height of a drawing object, drawing logic with a band raster, the start address of a corresponding drawing object, The amount of offset which skips this drawing object when a drawing object is drawn from the band middle (the number of Rhine), The start address (the link last is taken as NULL) of the continuing drawing information etc. is contained (each information should have only the number of bits which can express an assignment form in assignment resolution at least). Here, in this invention, although the class of the above-mentioned drawing object shall consist of a binary bit map, a multiple-value image, a rectangle, graphic data encoded by the run length format, a compression font pattern, and Rhine, if expressed as an intermediate form which can generate the bit image for carrying out printer en JINHE sending out, it will not specify especially the format of a drawing object itself.

[0048] In step S415 of drawing 4, for every band, the processing which usually draws the drawing object of a page and the font object contained in form reads the drawing information linked from the above-mentioned band table one by one, and draws a drawing object according to the printing positional information within each drawing information etc. Since it is usually generated ahead of a page object, drawing information is also linked previously and the font object in form is previously drawn as a result. Moreover, in step S410 of drawing 4, when generating a form raster, it becomes generable [which is considering as the completely same configuration as the above, and used the same drawing object drawing section 203c]. [generable] [it] However, the drawing information corresponding to each font pattern is not linking, and a font pattern is not drawn in a form raster by the font pattern generate time in form data.

[0049] Drawing 6 is a memory block diagram which is managed by 203d of form memory management sections and in which showing the outline of the memory configuration of 205d of form raster storing sections. Setting to this drawing, the managed table 601 for every form consists of information required in order to draw and identify each form. The storage region for the managed table 601 may be beforehand secured in a fixed area, and it may be constituted so that it may be secured dynamically, as possible. In the managed table 601, the start address of the form ID for identifying each form (it manages by the same number as a host computer), printing resolution, a division band total, the sense of a form, a bandwidth, and the form raster band table 602 is contained. The form raster band table 602 is a table which was divided in step S410 of drawing 4 and which stores the information on a band-like form raster unit, and only the number according to the number of bands in the managed table 601 is secured, and it contains the start address of each form raster, and the height of each band. In addition, as long as height other than the band of the lowest edge is fixed, said constant value and the lowest edge band height may be held in the form managed table 601. Moreover, the sense and bandwidth of resolution and a form are set among the information stored in the managed table 601 based on print data or panel default value, and Form ID is directed as a parameter of an overlay registration instruction by the

printer driver on a host etc.

[0050] Drawing 7 is the flow chart which showed the procedure of investigating whether the form raster corresponding to the form data with which 203d of form memory management sections was demanded being registered in the form raster storing section. That is, it is the procedure which showed the detail of step S408 of drawing 4. First, in step S701 of this drawing, it investigates whether it is in agreement with each of the form information which read the 1st form raster information and was specified by the form overlay run command from the form managed table 601 in 205d of form raster storing sections (steps S702–S706). That is, if not the same, in order to compare Form ID (step S702), to compare the following form information if the same, and to read the following form raster information, it progresses to step S708. Like the following the paper-size case by which the form conveyance direction of printing resolution (step S703), and a lengthwise direction/longitudinal direction is supported The sense of a form (step S704), In the case of a color printer, (step S706) is investigated for whether it is ***** without color mode (step S705) and page expansion / contraction assignment, respectively. When all the conditions are in agreement, a form number is memorized as form raster existence information at step S707, and it is judged that the form raster corresponding to an assignment form overlay is registered.

[0051] A form raster is judged [having not registered and] when retrieval of return and all form rasters is completed to step S701 on the other hand, if it investigates whether retrieval of all form rasters was completed and has not ended at step S708, when one information of steps S702–S706 is not in agreement (step S709).

<Generation of form raster from ROM data> drawing 8 is a flow chart which shows a form image generation procedure in case the form data of a Page Description Language format are storing ending in the non-volatile device (this device is hereafter explained as a flash ROM) of a flash ROM, a hard disk, etc. which can be written in. That is, when form data are offered by the flash ROM, a form raster is generated and held not only by the procedure of drawing 4 R> 4 but by the procedure of drawing 8.

[0052] Before performing at the idle time which has not received the print data which should be processed before receiving print data or usually receiving the print data of a page immediately after supplying a power source to an image output unit, processing of this flow chart may be constituted so that a form raster generation instruction may be received and executed from a host computer.

[0053] In this drawing, it investigates first whether it is equipped with the flash ROM at step S801, and when not mounted, processing is ended. On the other hand, when mounted, the printing conditions (printing resolution, the form conveyance direction, color mode, etc.) set up with the panel menu etc. at step S802 are read, and henceforth [continuing step S803], a form raster is generated from the form data memorized by the flash ROM, and it registers to the form raster storing section. First, it investigates whether the form data in a flash ROM are already registered into the form raster storing section (step S803), if it has not registered, form data will be read (step S804), the drawing object for 1 page containing a font pattern is generated (step S805), and each font pattern is registered to a font cache means.

[0054] When it investigates whether print data were received and print data exist in a receive buffer at continuing step S806, the drawing object [finishing / generation] corresponding to form data is deleted (step S814), and after releasing the work-piece memory which was being used as activity memory, form image generation processing is ended. Here, if it is in the middle of storing of a form image, the form image stored to the middle will also be released. In addition, with the gestalt of this operation, only in step S806, although existence of received data was checked, the step [not only this but] order which the processing time cuts in many, and the thing which do not use form by checking print data periodically and carrying out like this with a timer etc. and for which printing processing of data is usually barred are lost comparatively.

[0055] After drawing the form image for one band to a band buffer at continuing step S807 using said drawing objects other than a font object, said form image is agreement-ized at step S808. In continuing step S809, when it investigates whether memory space required for registration is in the form image storing section and **** memory space runs short, processing is ended, after performing an error message at step S810. On the other hand, if registration is possible, an agreement-ized finishing form image is stored (step S811). The drawing object which became

unnecessary [other than a font cache] at step S813 at the last is deleted repeatedly (step S812) until it all ends processing of the above-mentioned steps S806-S811 by 1 page, and processing of 1 form is ended. In addition, with the gestalt of this operation, although deed processing was interrupted for the error message at the time of out of memory, the priority of each form may be specified by panel setup etc., or it may constitute so that operating frequency may be memorized with form data, and it may delete from a form image storing field in the low order of priority, or order with low operating frequency, and processing of this form image storing may be continued.

[0056] With the above procedure, the form data which were registered by form data registration and which were described by the Page Description Language or the form data which is beforehand offered by ROM and which was described by the Page Description Language is changed into the drawing object by middle data with a form data generation instruction, and the raster data of the form are generated further, it encodes, and it is memorized. If it becomes clear that overlay is during delivery of the page which has received the delivery instruction at the page, the encoded form raster data which are memorized will be decoded, it will develop in band memory, and usual page objects other than form will be developed and printed on it.

[0057] Moreover, since it is held in the format developed by bit map data together with form data in a font cache about the alphabetic data which constitutes the form, about form data, the cache is carried out also including the alphabetic character contained in it, and it becomes possible to reuse.

[0058] By doing in this way, since the once registered form data are memorized as encoded raster data, it is not necessary to develop from a Page Description Language, and they can be quickly developed in band memory.

[0059] In addition, what is necessary is to only exclude the expansion processing to the raster data, to encode as it is and just to store it, when form data are transmitted not by the Page Description Language but by the image data.

[0060] With the gestalt of [gestalt of the 2nd operation] the 1st operation, since it registered with the font cache means fair [the alphabetic data contained in form data, and the alphabetic data usually used on a page], the font cache used within form may have been deleted by insufficient memory etc.

[0061] Therefore, it constitutes so that it may be hard to be deleted compared with the alphabetic data with which the font cache currently used within form data by setting up more highly

[priority] than the alphabetic data usually contained in a page to the font pattern generated and registered from the alphabetic data contained in the form data among the font patterns registered into the font cache means with the gestalt of this operation is usually used only within the page.

[0062] Drawing 9 is a block diagram explaining the basic configuration of the image output unit in which the gestalt of operation of the 2nd of this invention is shown, gives the same number about the same configuration as the gestalt of the 1st operation, and omits explanation. In this drawing, in case priority setup / judgment section 901 registers 203g HEFONTO pattern of font cache Management Department, it sets up priority, and it manages from which font pattern it deletes at the time of font pattern deletion. The configuration of those other than this is the same as that of drawing 2.

<Structure of font cache where priority was attached> drawing 10 R> 0 is the schematic diagram showing the configuration of the font cache stored in the font cache storing section in the gestalt of this operation.

[0063] In the gestalt of this operation, by using a character code as a search key, each font pattern 1006 follows indexes 1002 and 1003, and can search them now from the font information management table 501 (henceforth, managed table). Here, the character code given from the outside is 1 byte or 2 bytes, and an index has two steps of depth, the object (1002) for the high-order bytes of a character code, and the object (1003) for lower bytes. When registering a font pattern to a font cache, the pointer to the low order index or font cache stereo following the location corresponding to each of the high-order byte/lower byte of a character code is written in (initial value in an index is set to NULL). On the other hand, at the time of font cache retrieval, first, it investigates whether the pointer of the index for lower bytes is set to the location of the index corresponding to the high-order byte of said character code, and when set, a low order index is searched with the lower byte of a character code, and the pointer to the stereo of a font cache

is obtained similarly. In addition, the index for high-order bytes is constituted only from the member of code =0x00 by the coding scheme of a single cutting tool system.

[0064] In this drawing, the following information is stored in the managed table 1001 for every font as information for identifying each font. Namely, a typeface name, the graphic set in which allotment of a character code and a real pattern is shown, The flag which shows whether they are a character size and a scalable font, the stroke wait which shows the size of the line of an alphabetic character (un-illustrating), Others [flag / (un-illustrating) / the flag (un-illustrating) which shows a fixed pitch or a proportional pitch, and / which shows columnar writing/lateral writing], The pointer to the high order index 1002 for searching a font pattern etc. is stored, and the font number is assigned for every managed table (in drawing 10 , they are No. 1 and No. 2). In this drawing, the case where the double cutting tool system font is assigned and the font pattern corresponding to character code =0x2422 and 0x3021 is linked is shown in the font number =1 No. managed table.

[0065] The font cache 1004 consists of the header information section 1005 and a font pattern 1006. In addition to information required for drawing of the dot width of face w of a font pattern, dot height h, the start address of angle of rotation and a font pattern and the offset value from the base line, pitch information, etc., the font information used as the retrieval judging key of font caches, such as expansion and qualification information of the existence of board qualification, is set to the header information section 1005. Moreover, when the character pattern is compressed and stored, in addition to the above-mentioned information, it also has the agreement-sized flag which shows that it is compressed.

[0066] In the font cache which furthermore starts this invention, by having priority flag 1005a which shows whether this font cache was used in the fixed form form department, if it is during form data analysis, namely, if it is alphabetic data contained in form data, priority flag 1005a will be set 1 at font cache creation time, otherwise, 0 will be set (if it is the font cache usually created during page analysis). Moreover, when the font pattern which searches and asks for a font cache is found (it hit into the cache), it is during form data analysis, and when priority flag 1005a is set to 0, this priority flag is reset to 2. In addition, when priority flag 1005a is set to 2, it is shown that it is the font cache usually used from both pages in form.

[0067] On the other hand, when it hits to a font cache, it is usually during page data analysis, and also when the priority flag is set to 1, this priority flag is reset to 2, and it is shown that it is said font cache usually similarly used from both pages in form.

[0068] In addition, although drawing 10 explained the font pattern 1006 as a character pattern generated in RAM205, when it has a font pattern beforehand in a font ROM 204 like a dot font, a font pattern may not exist in RAM205. In this case, as for the start address of said font pattern in the header information section 1005, the address of the font pattern head in a font ROM 204 will be set.

[0069] In the font cache of a configuration of having explained above, the font cache which the font cache usually used only on the page is deleted by the 1st, and is continuously used only within form because a priority flag deletes in order of 0→1→2, and the font cache usually used for the last on both pages in form will be deleted at the times of font cache deletion, such as at the time of out of memory etc. When the font cache used in the fixed form form section becomes is hard to be deleted and it uses form especially by two or more pages with such a procedure, it becomes possible to raise the hit ratio of a font cache.

[0070] Drawing 11 is a flow chart which shows the expansion procedure of the font mentioned above. the memory space for developing a font — ** — in not being enough, it sets (step S111—YES) and a deletion threshold as 0 (step S112). Next, only the amount from which required memory space can secure the font cache which has the priority below the deletion threshold or all any, or little way is deleted (step S113). If it judges whether required memory was securable by that cause (step S114) and can secure, a font will be developed using the memory (step S115). Moreover, it judges (step S116), it will still be said that memory space is insufficient whether if memory is not enough, a current deletion threshold is 2, even if it will delete all font caches, if it is 2, and this considers it as an error. Moreover, if a deletion threshold is 0 or 1, one current deletion threshold will be raised and the font cache of the following priority will be deleted.

[0071] Thus, when becoming that it is hard to be deleted and using form especially by two or more

pages, the font cache used in the form section has the hit ratio of a font cache raised, and can raise the effectiveness of printing processing.

[0072] In addition, in order to show that priority 1 or the font cache of 2 is included for every class of font, the flag which indicates that the alphabetic character of a high priority is contained in the scalable flag column etc. may be formed. In this case, when a priority flag is set to 1 or 2, the flag which shows that the alphabetic character of this high priority (priority 1 or 2) is contained in coincidence is also set. Moreover, in deleting a font cache, first, for every class of font, whether the alphabetic character of a high priority is contained in the font judges, and if not contained, it deletes from the font. If it turns out that it is contained, other fonts will be investigated and the same processing will be repeated.

[0073] In addition, it does not matter even if it sets up the priority which deletes a font cache also using information, such as existence of qualification, font pattern size, and the 2nd level / level 1 kanji characters / the 1st level non-kanji character of JIS, in addition to the above-mentioned priority.

[0074]

[Other operation gestalten] In addition, even if it applies this invention to the system which consists of two or more devices (for example, a host computer, an interface device, a reader, a printer, etc.), it may be applied to the equipments (for example, a copying machine, facsimile apparatus, etc.) which consist of one device.

[0075] Moreover, the purpose of this invention supplies the storage which recorded the program code of the software which realizes the function of the operation gestalt mentioned above to a system or equipment, and is attained also by carrying out read-out activation of the program code with which the computer (or CPU and MPU) of the system or equipment was stored in the storage.

[0076] In this case, the function of the operation gestalt which the program code itself read from the storage mentioned above will be realized, and the storage which memorized that program code will constitute this invention.

[0077] As a storage for supplying a program code, a floppy disk, a hard disk, an optical disk, a magneto-optic disk, CD-ROM, CD-R, a magnetic tape, the memory card of a non-volatile, ROM, etc. can be used, for example.

[0078] Moreover, by performing the program code which the computer read, a part or all of processing that OS (operating system) which the function of the operation gestalt mentioned above is not only realized, but is working on a computer based on directions of the program code is actual is performed, and also when the function of the operation gestalt mentioned above by the processing is realized, it is contained.

[0079] Furthermore, after the program code read from the storage is written in the memory with which the functional expansion unit connected to the functional add-in board inserted in the computer or the computer is equipped, a part or all of processing that CPU with which the functional add-in board and functional expansion unit are equipped is actual performs, and also when the function of the operation gestalt mentioned above by the processing is realized, it is contained based on directions of the program code.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the sectional view of the laser beam printer which can apply this invention.

[Drawing 2] It is the block diagram of the image output unit shown in the gestalt of operation of the 1st of this invention.

[Drawing 3] It is the flow chart in the image output unit shown in the gestalt of the 1st operation which shows the outline of a form overlay activation procedure to form overlay registration processing, form raster registration processing, and a list.

[Drawing 4] It is the flow chart of the form overlay registration in the image output unit shown in the gestalt of the 1st operation, form raster registration, and form overlay executive operation.

[Drawing 5] It is the block diagram of the drawing object storing section of the image output unit shown in the gestalt of the 1st operation.

[Drawing 6] It is the block diagram of the form raster storing section of the image output unit shown in the gestalt of the 1st operation.

[Drawing 7] It is the flow chart of a form image retrieval procedure of the image output unit shown in the gestalt of the 1st operation.

[Drawing 8] It is the flow chart of the procedure of form raster generation and registration of the image output unit shown in the gestalt of the 1st operation in case form data are stored in nonvolatile memory.

[Drawing 9] It is the block diagram of the image output unit shown in the gestalt of operation of the 2nd of this invention.

[Drawing 10] It is the block diagram of the font cache storing section in the image output unit shown in the gestalt of the 2nd operation.

[Drawing 11] It is the flow chart of font expansion processing of the image output unit shown in the gestalt of the 2nd operation.

[Description of Notations]

100 Airline Printer

201 CPU

203 Program ROM

203a Drawing object generation section

203b Overlay registration / read-out section

203d Form memory management section

203e Form raster drawing section

203f The formation of raster agreement, and the decryption section

203g Font cache Management Department

205 RAM

205a Drawing object storing memory

205b Form data storage section

205c Page memory (band raster)

205d Form raster storing memory

207 Receive Buffer

501 Page Information Header Unit

502 Overlay Existence Information

503 Band Table
504 Drawing Information Storing Memory
505 Drawing Object Storage Section
601 Form Raster Managed Table
602 Form Band Table
603 Agreement-sized Form Raster
901 Priority Setup / Judgment Section
1001 Font Information Management Table
1002-1003 Index
1004 Font Cache
1005 Font Cache Header Information Bureau
1006 Font Pattern

[Translation done.]

* NOTICES *

JPO and NCIP are not responsible for any damages caused by the use of this translation.

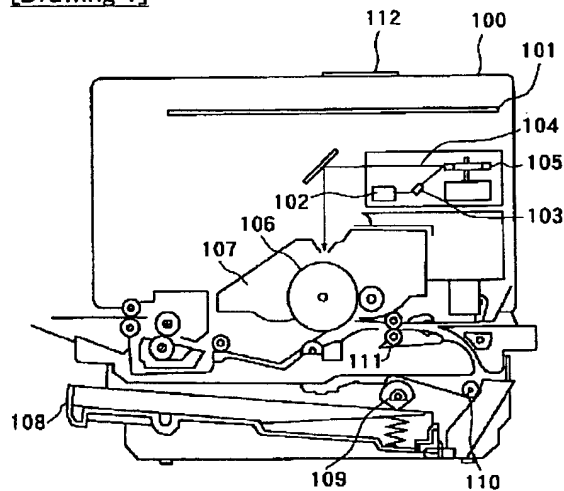
1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

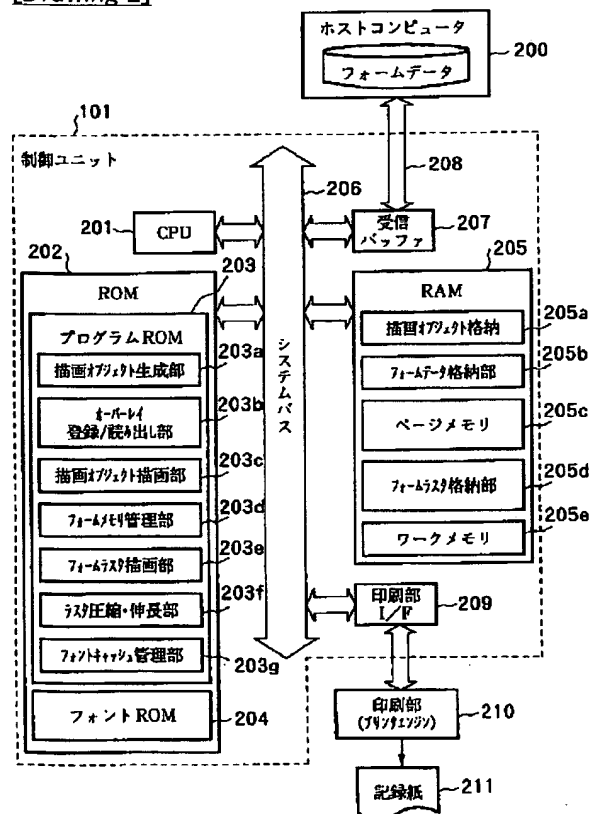
3.In the drawings, any words are not translated.

DRAWINGS

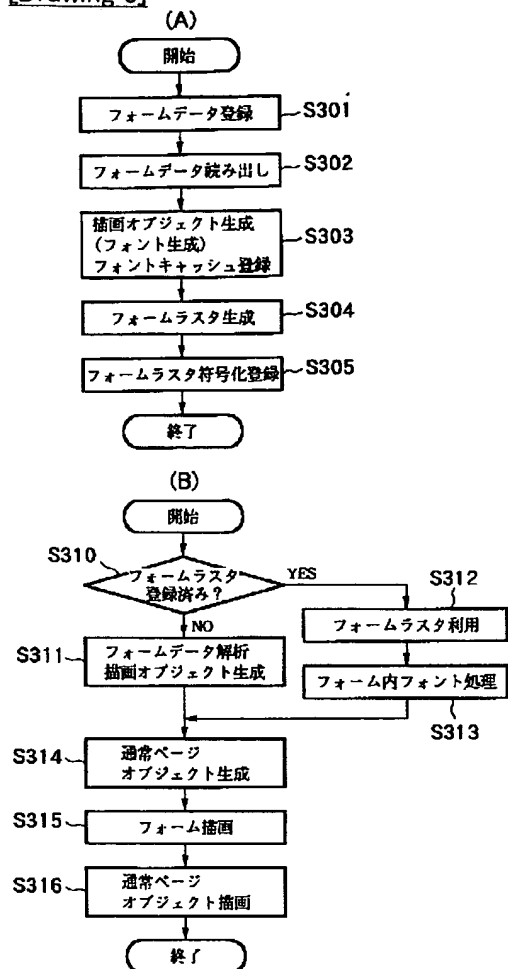
[Drawing 1]



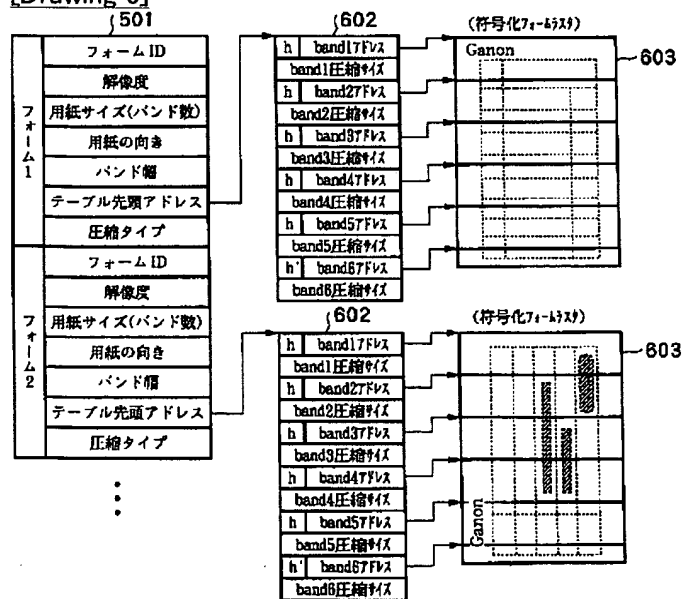
[Drawing 2]



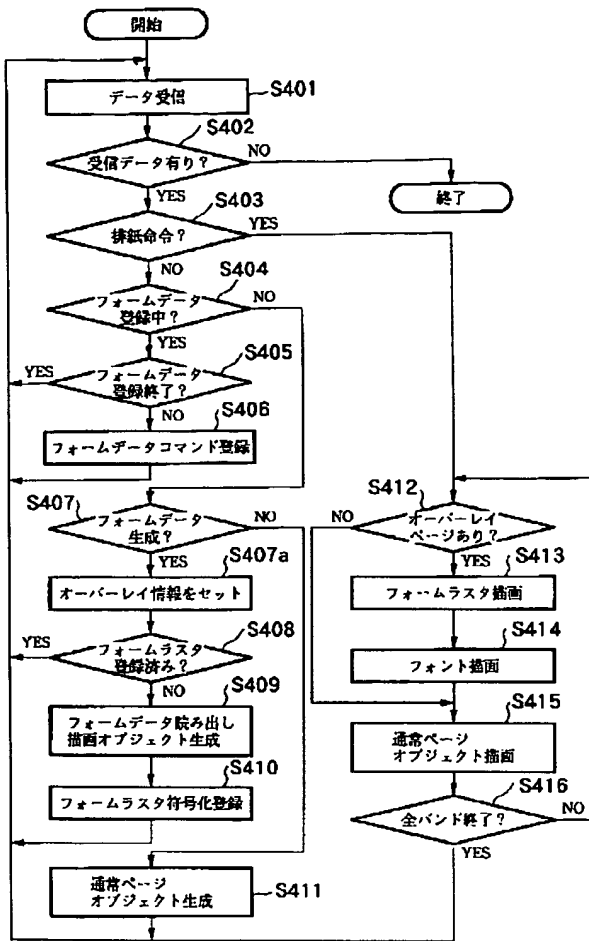
[Drawing 3]



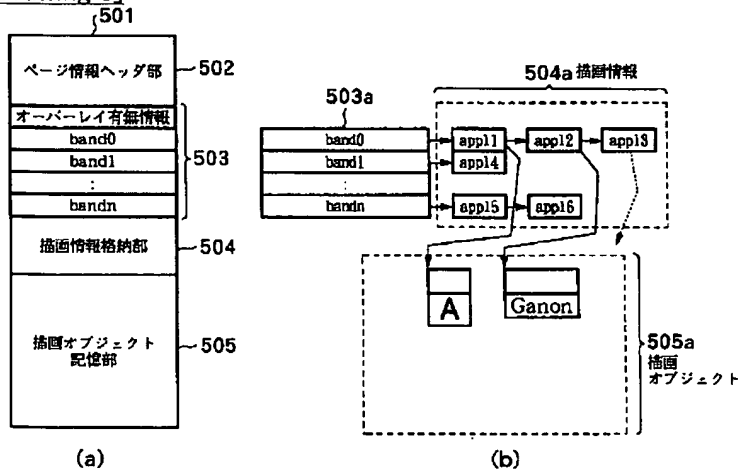
[Drawing 6]



[Drawing 4]



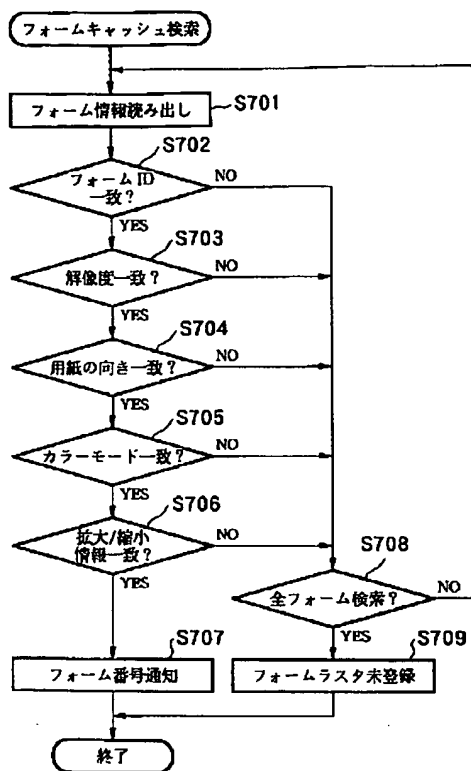
[Drawing 5]



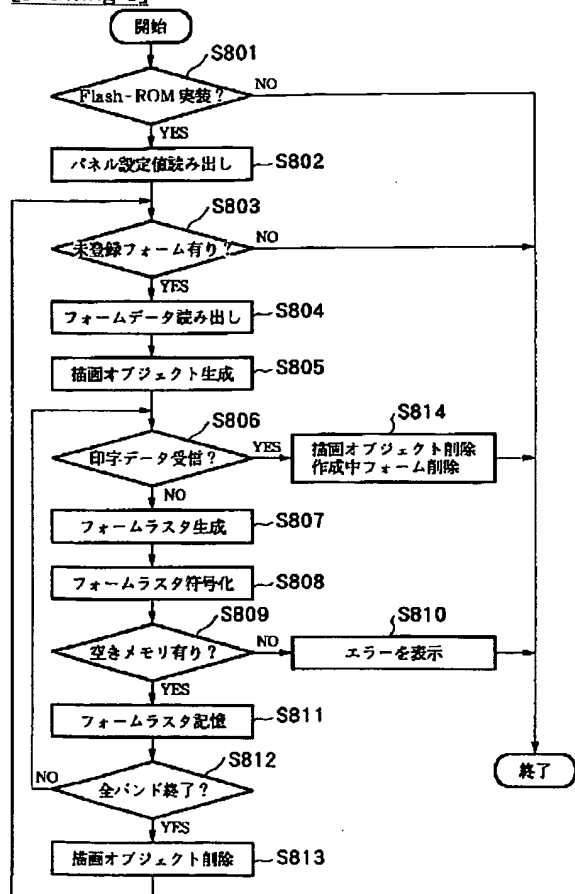
アプリの種類	アプリ内 印字位置	描画高さ	描画論理	アプリの 先頭アドレス	アプリ のセクタ	次の描画 情報部のアドレス
--------	--------------	------	------	----------------	-------------	------------------

(c)

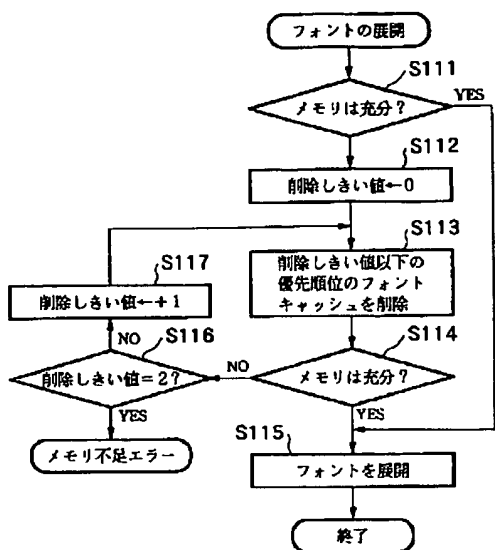
[Drawing 7]



[Drawing 8]



[Drawing 9]



[Translation done.]

(11)特許出願公開番号

(43)公開日 平成11年(1999)6月29日

A
Z

(74)代理人 弁理士 大塚 康徳 (外2名)

```

graph TD
    Start([開始]) --> S401[データ受信]
    S401 --> S402{受信データ有り?}
    S402 -- YES --> S403{画面命令?}
    S402 -- NO --> End([終了])
    S403 -- YES --> S404{フォームデータ登録中?}
    S403 -- NO --> S404
    S404 -- YES --> S405{フォームデータ登録終了?}
    S404 -- NO --> S405
    S405 -- YES --> S406[フォームデータコマンド登録]
    S405 -- NO --> S406
    S406 --> S407{フォームデータ登録?}
    S407 -- YES --> S407a[オーバーレイ情報をセット]
    S407 -- NO --> S407a
    S407a --> S408{フォームラスタ登録終了?}
    S408 -- YES --> S409[フォームデータ読み出し  
画面オブジェクト生成]
    S408 -- NO --> S409
    S409 --> S410[フォームラスタ符号化登録]
    S410 --> S411[通常ページ  
オブジェクト生成]
    S411 --> S412{オーバーレイ  
ページあり?}
    S412 -- YES --> S413[フォームラスタ描画]
    S412 -- NO --> S415[通常ページ  
オブジェクト描画]
    S413 --> S414[フォント描画]
    S414 --> S415
    S415 --> S416{全ページ終了?}
    S416 -- YES --> End
    S416 -- NO --> S412
  
```

【特許請求の範囲】

【請求項1】 文字を含まないビットマップデータとして生成されたフォームデータを符号化して格納するフォーム格納手段と、

フォームデータをビットマップデータとして生成し、符号化して前記フォーム格納手段に格納するフォーム生成手段と、

ビットマップデータとして生成された文字データを格納するフォントキャッシュと、

文字データをビットマップデータとして生成し、前記フォントキャッシュに格納する文字生成手段と、

印刷データをフォームデータとオーバーレイして印刷する際に、格納された符号化されたフォームデータを復号するとともに、フォームデータに含まれる文字データは前記フォントキャッシュに格納されたビットマップデータを用いて描画を行い、前記印刷データを重ねて出力するオーバーレイ手段とを備えることを特徴とする画像出力装置。

【請求項2】 前記フォーム生成手段は、印刷データとともに受信するフォームデータを基に、ラスタデータを生成して符号化し格納することを特徴とする請求項1に記載の画像出力装置。

【請求項3】 前記フォーム生成手段は、予め不揮発性メモリに格納されたフォームデータを基に、ラスタデータを生成して符号化し格納することを特徴とする請求項1に記載の画像出力装置。

【請求項4】 前記フォーム生成手段は、電源投入時あるいは空き時間に、フォームデータを読み出してフォーム画像を生成することを特徴とする請求項3に記載の画像出力装置。

【請求項5】 前記フォーム生成手段は、ホストから指示されるフォーム生成命令によってフォーム画像を生成することを特徴とする請求項3に記載の画像出力装置。

【請求項6】 前記フォーム生成手段は、1ページを分割してなるバンド単位でラスタデータを生成し、符号化して格納することを特徴とする請求項1に記載の画像出力装置。

【請求項7】 前記フォーム生成手段は、1ページ単位でラスタデータを生成し、符号化して格納することを特徴とする請求項1に記載の画像出力装置。

【請求項8】 前記オーバーレイ出力手段は、フォームデータ以外の通常ページデータを、フォームデータの上に重複させることを特徴とする請求項1に記載の画像出力装置。

【請求項9】 前記文字生成手段は、フォームデータに含まれる文字データを含む文字データをイメージデータとして生成して前記フォントキャッシュに格納することを特徴とする請求項1に記載の画像出力装置。

【請求項10】 前記文字生成手段は、フォームデータに含まれる文字データについては、通常ページデータに

含まれる文字データよりも高い優先順位を設定してフォントキャッシュに格納し、文字データをビットマップに展開する際にフォントキャッシュに格納されたデータを削除する場合には、優先順位の低い文字データから削除することを特徴とする請求項1に記載の画像出力装置。

【請求項11】 文字を含まないフォームデータをビットマップデータとして生成し、符号化してフォーム格納手段に格納するフォーム生成工程と、

文字データをビットマップデータとして生成し、フォントキャッシュに格納する文字生成工程と、

印刷データをフォームデータとオーバーレイして印刷する際に、格納された符号化されたフォームデータを復号するとともに、フォームデータに含まれる文字データは前記フォントキャッシュに格納されたビットマップデータを用いて描画を行い、前記印刷データを重ねて出力するオーバーレイ工程とを備えることを特徴とする画像出力方法。

【請求項12】 前記フォーム生成工程は、印刷データとともに受信するフォームデータを基に、ラスタデータを生成して符号化し格納することを特徴とする請求項11に記載の画像出力方法。

【請求項13】 前記フォーム生成工程は、予め不揮発性メモリに格納されたフォームデータを基に、ラスタデータを生成して符号化し格納することを特徴とする請求項12に記載の画像出力方法。

【請求項14】 前記フォーム生成工程は、電源投入時あるいは空き時間に、フォームデータを読み出してフォーム画像を生成することを特徴とする請求項13に記載の画像出力方法。

【請求項15】 前記フォーム生成工程は、ホストから指示されるフォーム生成命令によってフォーム画像を生成することを特徴とする請求項13に記載の画像出力方法。

【請求項16】 前記フォーム生成工程は、1ページを分割してなるバンド単位でラスタデータを生成し、符号化して格納することを特徴とする請求項11に記載の画像出力方法。

【請求項17】 前記フォーム生成工程は、1ページ単位でラスタデータを生成し、符号化して格納することを特徴とする請求項11に記載の画像出力方法。

【請求項18】 前記オーバーレイ出力工程は、フォームデータ以外の通常ページデータを、フォームデータの上に重複させることを特徴とする請求項11に記載の画像出力方法。

【請求項19】 前記文字生成工程は、フォームデータに含まれる文字データについては、通常ページデータに含まれる文字データよりも高い優先順位を設定してフォントキャッシュに格納し、文字データをビットマップに展開する際にフォントキャッシュに格納されたデータを削除する場合には、優先順位の低い文字データから削除

することを特徴とする請求項 1 に記載の画像出力方法。

【請求項 20】 文字を含まないフォームデータをラスターデータとして生成し、符号化してフォーム格納手段に格納するフォーム生成手段と、

文字データをビットマップデータとして生成し、フォントキャッシュに格納する文字生成手段と、

印刷データをフォームデータとオーバーレイして印刷する際に、格納された符号化されたフォームデータを復号するとともに、フォームデータに含まれる文字データは前記フォントキャッシュに格納されたビットマップデータを用いて描画を行い、前記印刷データを重ねて出力するオーバーレイ手段を有する装置としてコンピュータに機能させるプログラムを格納したことを特徴とするコンピュータ可読メモリ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えば通常ページデータとフォームデータをオーバーレイ出力可能な画像出力装置及び方法、特にレーザビームプリンタ等のプリンタに関する。

【0002】

【従来の技術】一般的な業務上の印刷においては、帳票や定型文書のように、複数ページにわたって同一内容を印字する定型フォーム部（タイトルや枠線など）と、ページ毎に内容が異なる非定型データ部（数字や名前など。通常ページデータと呼ぶ）を重ね合わせて印刷する、フォームオーバーレイ印刷が用いられる場合が多い。

【0003】そこで、ページ記述言語を解析しラスター画像を生成する従来のページプリンタにおいては、前記ページ記述言語形式で作成した定型フォーム部をイベント駆動マクロ命令としてプリンタの記憶領域に登録しておき、排紙動作等のイベントによって前記マクロを自動実行することで、フォームオーバーレイ印刷を実現していた。前記イベント駆動マクロは、該マクロの指定時（1 回目の実行）と、1 ページ分の通常ページデータの解析が終了した後、例えば排紙命令を受信する度（2 回目以降）に登録内容が読み出され、ページ記述言語解析手段によって通常ページデータと同様に解析され、1 ページ分のラスター画像を生成するための中間コード（以下、描画オブジェクトと呼ぶ）に変換される。

【0004】また、印刷実行に先だって、予めホストコンピュータからプリンタへ前記フォームデータ（マクロデータ）を転送し、フラッシュメモリ等の不揮発性メモリへ登録しておくことで、毎回フォームデータを転送しなくてもよいように構成されている場合もある。この場合、印刷実行時には、フラッシュメモリ内のフォームデータをイベント駆動マクロとして指示する命令のみが送られることになる。

【0005】

【発明が解決しようする課題】しかしながら、上記従来例では、以下に説明するような問題点があった。

【0006】すなわち、前記イベント駆動マクロによってフォームオーバーレイ印刷を実現している場合、フォームデータをページ記述言語形式で登録しているため、複雑なフォームデータの場合は、フォームデータそのものの解析と描画オブジェクトの生成に時間がかかってしまっていた。すなわち、1 ページ目の出力は、オーバーレイを用いずに通常データとして同様の内容を出力する場合と同等の処理速度しか得られず、また、フォーム内容が複雑になればなるほど、フォーム部の処理時間が増大していた。

【0007】また、同一フォームを複数ページで使用している印刷データを出力する際、ページ毎に毎回、同一のマクロ（コマンド群）を解析し、同一の描画オブジェクトを生成するため、同一内容のページであるにも関わらず、2 ページ目以降の印字速度が向上しないという問題点があった。すなわち、同一の描画オブジェクトを、毎ページ生成するため、複数ページ間で描画オブジェクトの共有ができなかった。そのため、複数ページの描画処理に必要なメモリが増大し、排紙待ちが発生する等のパフォーマンス低下も懸念されていた。特に、図形やイメージ画像を多く含む複雑なフォームをオーバーレイを出力指定した際には、描画メモリ不足などのエラーも発生しやすくなり、上記課題の解決が重要となっていた。

【0008】上記問題の解決策の一つとしては、予めフォームデータをラスターライズし、イメージデータ（フォーム画像）としてプリンタ内に保持しておき、通常ページデータと重ね合わせて描画することが考えられる。しかし、1 ページ分の画像を保持するためには大量のメモリが必要となり、実用的ではない（A4、600 dpi、モノクロで約 8 MB のメモリが必要）。

【0009】そこで、フォーム画像を符合化（圧縮）することが考えられるが、フォーム画像の圧縮に適応可能なパケット圧縮（バイトオーダーの連続性に注目して画像を符合化する）などは、文字を含むパターンは圧縮率が一般的に良くないため、フォームデータに文字データが多く含まれる場合は、フォーム画像の圧縮率が低下してしまうという問題があった。

【0010】本発明の上記問題点に鑑みてなされたもので、フォーム内容が複雑になってもフォーム部の処理時間を増大させず、また、同一内容のフォームであれば、2 ページ目以降の印字速度を向上させることができ、また、複数ページの描画処理のために必要なメモリを増大させず、排紙待ち等のパフォーマンスを低下させない画像出力装置及び方法を提供することを目的とする。

【0011】

【課題を解決するための手段】上記問題点を解決するための本発明は以下のような構成を備える。即ち、文字を

含まないビットマップデータとして生成されたフォームデータを符号化して格納するフォーム格納手段と、フォームデータをビットマップデータとして生成し、符号化して前記フォーム格納手段に格納するフォーム生成手段と、ビットマップデータとして生成された文字データを格納するフォントキャッシュと、文字データをビットマップデータとして生成し、前記フォントキャッシュに格納する文字生成手段と、印刷データをフォームデータとオーバーレイして印刷する際に、格納された符号化されたフォームデータを復号するとともに、フォームデータに含まれる文字データは前記フォントキャッシュに格納されたビットマップデータを用いて描画を行い、前記印刷データを重ねて出力するオーバーレイ手段とを備える画像出力装置。

【0012】あるいは、文字を含まないフォームデータをビットマップデータとして生成し、符号化してフォーム格納手段に格納するフォーム生成工程と、文字データをビットマップデータとして生成し、フォントキャッシュに格納する文字生成工程と、印刷データをフォームデータとオーバーレイして印刷する際に、格納された符号化されたフォームデータを復号するとともに、フォームデータに含まれる文字データは前記フォントキャッシュに格納されたビットマップデータを用いて描画を行い、前記印刷データを重ねて出力するオーバーレイ工程とを備える画像出力方法。

【0013】あるいは、文字を含まないフォームデータをラスターデータとして生成し、符号化してフォーム格納手段に格納するフォーム生成手段と、文字データをビットマップデータとして生成し、フォントキャッシュに格納する文字生成手段と、印刷データをフォームデータとオーバーレイして印刷する際に、格納された符号化されたフォームデータを復号するとともに、フォームデータに含まれる文字データは前記フォントキャッシュに格納されたビットマップデータを用いて描画を行い、前記印刷データを重ねて出力するオーバーレイ手段を有する装置としてコンピュータに機能させるプログラムを格納したコンピュータ可読メモリ。

【0014】

【発明の実施の形態】本実施形態の構成を説明する前に、本発明を適用するに好適な画像処理装置の構成について、図1を参照しながら説明する。図1は本発明を適用可能な画像処理装置の構成を示す断面図であり、例えば、レーザービームプリンタ（以下LBP）の場合を示す。なお、本発明を適用可能な画像処理装置はLBPに限られるものではなく、他のプリント方式のプリンタ装置であっても良いことはいうまでもない。

【0015】同図において、LBP本体100は、外部に接続されているホストコンピュータ（図2の200）からページ記述言語の形式で供給される文字情報（文字コード等）や図形情報や、定型書式データ（フォームデ

ータ）あるいは他のマクロ命令などを入力して記憶するとともに、それらの情報に従って対応する文字パターンや図形パターンなどを作成し、記録媒体である記録紙上に像を形成する。操作パネル112は、操作のためのスイッチおよびLED表示器などが配されている。プリンタ制御ユニット101は、LBP100全体の制御およびホストコンピュータから供給される文字情報などを解析する。この制御ユニット101は、主に文字情報に対応する文字パターンのビデオ信号に変換してレーザードライバ102に出力する。また、このLBPは不図示のデータ源から文字パターンの登録やフォームデータなどの登録も行え、同様に印字データに応じてビデオ信号に変換してレーザードライバ102に出力する。レーザードライバ102は半導体レーザー103を駆動するための回路であり、入力されたビデオ信号に応じて半導体レーザー103から発射されるレーザー光104をオンオフ切り替える。レーザー104は回転多面鏡105で左右方向に振られ静電ドラム106上を走査する。これにより、静電ドラム106上には文字パターンの静電潜像が形成される。この潜像は、静電ドラム106周囲の現像ユニット107により現像された後、記録紙に転送される。この記録紙にはカットシートを用い、カットシート記録紙はLBP100に装着した用紙カセット108に収納され、給紙ローラ109および搬送ローラ110と111とにより装置内に取り込まれて、静電ドラム106に供給される。

【0016】〔第1の実施の形態〕以下に添付の図面を参照して、本発明の第1の実施の形態について説明する。

【0017】図2は本発明の第1の実施の形態を示す画像出力装置の基本構成（図1の画像処理装置の制御ユニット101の内部構成）を説明するブロック図であり、例えばレーザービームプリンタの場合を示している。

【0018】また、本発明の機能が実行されるのであれば、単体の機器であっても、LAN等のネットワークを介して処理が行なわれるシステムであっても良い。

【0019】同図において、プリンタ本体100は、所定のインターフェイス208を介して外部のホストコンピュータ200と接続されており、ホストコンピュータ200から送られる印刷データに従って所定の印刷処理を実行するように構成されている。ホストコンピュータ200から受信した該印刷データは受信バッファ207に受信データとして一時的に蓄えられる。

【0020】なお、ホストコンピュータ200のハードディスク等には、ページ記述言語形式で記述されたフォームデータが格納されており、個々のフォームデータは一意に識別可能なフォームID等の情報と対応付けて管理されている。また、フォームオーバーレイ印刷時は、前記フォームデータはフォーム登録命令（イベント駆動マクロ）として送出される。

【0021】プリンタCPU201は、読み出し専用メモリであるROM202に記憶された制御プログラム203に基づいてシステムバス206に接続された各種デバイスとのアクセスを統括的に制御し、受信バッファ207に蓄えられた印刷データを解析し、少なくとも1ページ分の描画オブジェクトを生成し（描画オブジェクト生成部203a）、続いて前記描画オブジェクトを元に1ページ分あるいは複数のバンドに分割した1バンド分のラスティメージを生成し（描画オブジェクト描画部203c）、印刷部I/F・209を介して印刷部210（プリンタエンジン）へ、前記ラスティメージのビットのON/OFFによるビデオ信号を出力する（シップする）。印刷部210はCPU201から受け取ったビデオ信号を元に記録紙211上に画像の印刷を行なう。ここで、描画オブジェクト生成部203aには、文字データ（書体、サイズ、文字コード）に対応するフォントパターンを生成するフォントスケラも含まれる。

【0022】なお、ラスティメージを格納するメモリを1ページ分用意せずに、ページをバンド状に分割した少なくとも2バンド分を用意する場合（以下、バンドラスタ）は、前記ラスティメージの生成とシップ動作を同時に行なうよう構成されている。すなわち、描画済みの一方のバンドラスタをシップしながら、他方のバンドラスタに属する描画オブジェクトを描画する（以下、上記描画手順をバンディングによる描画と呼ぶ）。

【0023】また、制御プログラム203は、上記構成以外に、以下の機能を実現するためのプログラムから構成されている。

【0024】すなわち、受信バッファ207内の印刷データがフォーム登録命令であった場合に、ページ記述言語で記述された前記フォームデータをRAM205（後述）内へ登録し、フォームオーバーレイ実行命令受信時に、前記フォームデータを読み出すオーバーレイ登録／読み出し部203bと、読み出したフォームデータを元に文字を含まないフォームラスタを生成し、文字データはフォントパターンを生成する描画オブジェクト描画部203cと、前記フォームラスタをRAM205内に記憶・保持するフォームメモリ管理部203dと、前記フォントパターンをフォントキャッシュ格納部へ登録し、同一文字データの出力が要求された場合はフォントスケラを用いずに登録済みのフォントパターンを再利用するフォントキャッシュ管理部203gと、前記フォームラスタを記憶する前に符合化し、通常ページと重ね合わせて描画する前に復号化するフォーム圧縮・伸長部203fと、前記復号化済みのフォームラスタを通常ページデータと重ね合わせてページメモリ205cへ描画するフォームラスタ描画部203eとから構成されている。

【0025】なお、前記フォームメモリ管理部203dは、要求されたフォームデータに対応するフォームラスタがフォームラスタ格納部205dに記憶済みかどうか

調べ、既に記憶済みであった場合は、記憶済みフォームラスタを用いるキャッシュ機能を備える（ただし、フォームデータ中の文字データに関して、生成済みのフォントキャッシュを利用するために、オーバーレイ格納部203aからフォームデータの読み出しは行なう）。また、このROM202は、文字出力に用いるドットフォントやスケラブルフォントから構成されるフォントデータを格納するメモリとしても用いられている（フォントROM204）。

【0026】RAM205は、CPU201の主メモリ、ワークメモリ等として機能するRAMであり、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張できるように構成されている。RAM205は、前記描画オブジェクトを格納する描画オブジェクト格納部205a（前記フォントメモリ204内のスケラブルデータに基づいて展開された文字パターンをキャッシュするフォントキャッシュ登録メモリを含む）や、制御プログラム203aによって一時的に使用されるワークメモリ205e、1ページをバンド状に分割したバンド領域の2面に相当するラスティメージ（あるいは1ページ分のラスティメージ）を格納するページメモリ205c、ページ記述言語で記述されたフォームデータを格納するフォームデータ格納部205b、前記フォームデータを解析して得られたフォーム画像を符合化して保持するためのフォームラスタ格納部205dの他、印刷環境データ格納メモリ等（不図示）に用いられる。なお、LBP100は図示しない電源部から電力の供給を受けている。また、上記ではフォームデータはRAM205に格納すると説明したが、予めフラッシュメモリ等の不揮発性メモリに格納しておいても構わない。

【0027】なお、本発明を構成する装置が、ROMに格納されたプログラムとして供給されるよう説明したが、これに限らず、フロッピーディスクやハードディスク等の媒体によって供給され、実行前にRAM等へロードされた後、実行されるよう構成されていても構わない。

<オーバーレイ出力処理手順>続いて、上記構成による画像出力装置における、オーバーレイ出力の処理手順を、図3、図4のフローチャートを用いて以下に説明する。

【0028】<フォームデータ登録の概要>図3（A）は、ホストコンピュータ200よりフォームデータを受信した際に、フォームラスタを生成・登録するための命令手順の概略を示したフローチャートである。

【0029】まず、ステップS301において、フォーム登録命令を受信したら、フォームデータをプリンタ内のフォームデータ格納部205bへ登録する。ここで、フォーム登録命令とは、オーバーレイ登録開始命令～フォームの実データ～オーバーレイ登録終了命令から構成

されている一連の命令群を意味する。続いて、フォームラスタ生成命令に従い、ステップS301にて登録したフォームデータを読み出し（ステップS302）、フォームデータを解析し、対応する1ページ分のフォントパターンを含む描画オブジェクトを生成する（ステップS303）。なお、生成されたフォントパターンは、通常ページの文字データと同様、フォントキャッシュへ登録しておく。

【0030】続いて、フォントオブジェクト以外の描画オブジェクトをバンドラスタへ描画し、フォームラスタを生成する（ステップS304）。フォームラスタの生成が終了したら、ステップS305にて前記フォームラスタを符合化した後、フォームを識別するためのフォームIDなどの情報と共にフォームラスタを登録して、フォームラスタ登録処理を終了する。

【0031】＜フォームオーバーレイ実行の概要＞図3(B)は、フォームオーバーレイ実行命令を受信した場合の処理手順の概略を示したフローチャートである。なお、図3(A)に示したフォームデータの登録と、図3(B)のフォームオーバーレイ実行とは連続して（同一の印刷ジョブで）受信・処理しても良いし、フォームデータの登録のみを先に行なっても構わない。

【0032】同図において、まず、ステップS310において、フォームオーバーレイ実行命令で指定されたフォームIDに対応するフォームラスタが登録済みかどうかを調べる（フォームラスタ管理部203d）。既に登録されている場合は、該フォームラスタを描画する指示をフォームラスタ描画部203eへ伝え（ステップS312）、同時にフォームデータの文字データ部分のみを解析し、フォントオブジェクトの生成を行なう（ステップS313）。この時、フォントオブジェクトは図3(A)のステップS303で登録したフォントキャッシュにヒットするため、スケーラによるスケーリング処理は行なわれない。

【0033】一方、ステップS310にて所望のフォームラスタが未登録であると判断された場合は、ステップS311にて、フォームデータを全て読み出し、描画オブジェクトを生成する。描画オブジェクトの生成自体は、フォームラスタ登録時と同様に行なえば良い。

【0034】上記のようにフォームデータの処理が終了した後、非定型部の通常ページデータを解析し、ステップS311などと同様に描画オブジェクトを生成する（S314）。

【0035】1ページ分のフォームデータと通常ページデータを処理した後、フォーム部分をバンドラスタへ描画する（ステップS315）。すなわち、フォームラスタが登録済みの場合は、符合化フォームラスタとフォームに含まれるフォントオブジェクトを描画し、一方、未登録だった場合は、フォーム部分に相当する描画オブジェクト群（フォントオブジェクトを含む）をバンドラスタ

タへ描画する。最後に、通常ページの描画オブジェクトを同様に描画し（ステップS316）、フォームオーバーレイの実行処理を終了する。

【0036】さて、図3に示したフローチャートは、フォームの登録を示す手順と、フォームオーバーレイの実行を示す手順とを、解りやすくするために独立して示したものであるが、実際には、ホストコンピュータから受信する命令を解析し、コマンドに応じてこれらの処理は進められることになる。そのコマンド解析まで含めた処理手順が図4に示したフローチャートとである。

【0037】＜ページオブジェクトの生成処理＞図4は、フォームラスタの生成・登録、およびフォームオーバーレイの実行を含めた頁オブジェクトの生成処理手順を示したフローチャートである。

【0038】同図において、まず制御プログラム203は、ステップ401にて、ホストコンピュータ200から送られる印刷データを受信バッファ207へ記憶する。受信バッファ207内に印刷データが存在するかどうかを調べ（ステップS402）、存在しない場合は処理を終了する。一方、印刷データが存在する場合は、続くステップS403以下で印刷データの解析を行なう。まずステップS403にて、処理中の印刷データが排紙命令かどうかを調べ、排紙命令でないと判断した場合には、続くステップS404以下で印刷データの解析を続ける。なお、ステップS403においては、簡単のため排紙命令（フォームフィードコマンド）かどうかを調べると説明したが、実際には、文字印字位置移動命令等で、印字中のページ下端を越えるような、排紙動作を引き起こす要因となりうる印字データの場合も、同様にステップS408以降へ進む。

【0039】ステップS404では、前記印刷データがオーバーレイ登録開始命令かあるいはオーバーレイ登録中であるかどうかを調べ、そうであった場合は、オーバーレイ登録終了命令を受信するまで（ステップS405）、フォームデータをフォームデータ格納部205bへ書き込み（ステップS406）、続く印刷データを受信バッファ207より読み出すためにステップS401へ戻る。なお、ステップS404において、フォームデータ登録中であるかどうかを判断するためには、電源投入時等に0にリセットされるフラグを用意しておき、オーバーレイ登録開始命令受信時に、前記フラグを1にセットし、前記フラグがセットされている場合は、フォームデータ登録中であると判断し、オーバーレイ登録終了命令受信時（S405）に前記フラグをリセットする。あるいは、オーバーレイ登録開始命令のパラメータとして登録すべきフォームデータのサイズをカウンタとして受けとり、前記カウンタが0になるまでフォームデータを登録するように構成しても構わない。なお、オーバーレイ格納部として、フラッシュメモリ等の不揮発性メモリを設け、印刷データ受信前に予めフォームデータを登

録しておくように構成した場合は、ステップS404、ステップS406の各ステップは実行しなくても構わない。

【0040】ステップS404において、フォームデータ登録中でないと判断した場合は、続くステップS407において、該印刷データがフォームオーバーレイ実行命令かどうかを調べ、フォームオーバーレイ実行命令であると判断された場合は、処理中の通常ページに重ね合わせるべきフォームデータが存在することを示す情報

(後述)を描画オブジェクト格納部205aへセットした後、フォームデータに対応するフォームラスタがフォームラスタ格納部205dに登録済みかどうかを調べる(ステップS408)。未登録であった場合は、ステップS409にて、オーバーレイ格納部205aより前記フォームデータを読み出し、描画オブジェクト生成部203aによって、フォームデータ中の各印刷データ(文字データを含む)に対応する描画オブジェクトを生成する(ステップS409)。

【0041】なお、ステップS409における描画オブジェクトの生成は、通常ページに印字されるデータの解析および描画オブジェクトの生成と同様の処理で良いが(ステップS411にて後述)、フォントパターン以外の描画オブジェクトは、通常ページの描画オブジェクトとは別領域(例えば、ワークメモリ205d)に格納されるものとし、フォントパターンはフォントキャッシュ領域に登録される。なお、フォントキャッシュに登録された文字データは、通常ページデータの処理時にも使用され、キャッシュにヒットする。また、ステップS409においては、描画オブジェクトの生成と同時に該フォームを識別するための管理情報もあわせてワークメモリへ一時的に記憶しておく。ここで前記管理情報には、各フォーム毎に一意に決まるフォームID(ホストより指定される)、用紙サイズ、用紙の印字方向、データ生成解像度等が含まれる。

【0042】続くステップS410において、フォント以外の描画オブジェクトを描画してフォームラスタを生成した後(描画オブジェクト描画部203c)、フォーム画像を符合化し(ラスタ符合化部203f)、さらに前記符合化済みフォーム画像をフォームラスタ格納部205dへ格納する(フォームメモリ管理部203d)。すなわち、S409にて生成した前記フォームデータ対応の各描画オブジェクト(フォントパターンを除く)を、一時的にページメモリ205c上へ描画した後、フォームラスタ格納部205dへ格納し、最後にワークメモリに生成した描画オブジェクトをクリアする。本実施の形態においては、この時生成するフォームラスタは、1ページを複数のバンドに分割した状態で生成・格納するものとし、なおかつ分割高さは、ページメモリ(バンドラスタ)205cが取り得る最小高さとなるように構成する。このように構成することで、搭載メモリによ

てバンドラスタ205cが256ドット/512ドット/1024ドット高さの3段階を取り得るような場合でも、フォームラスタは256ドット高さで生成・格納するため、異なる条件下で同一フォームラスタを使用することができる。なお、フォームラスタへ描画オブジェクトを描画する手順は、シップ時に通常ページの描画オブジェクトをラスタメモリへ描画する手順と全く同様に行なえばよい。

【0043】一方、ステップS407にて、印刷データがフォームオーバーレイ実行命令でないと判断された場合は、通常ページに描画する印刷データとして解析し、描画オブジェクトを生成する(ステップS411)。すなわち、印刷データが文字印字命令であった場合は、印刷データで指定された文字情報に対応するフォント情報をフォントROM204より読み出し、対応するドットパターン(=描画オブジェクト)を生成し、印字位置情報や幅、高さ、描画論理等、ページラスタへ描画する際に必要な情報と共に、前記ドットパターンを描画オブジェクト格納部205aへ格納する(文字以外の図形データやイメージデータであった場合も、同様に各印字命令に対応した中間コードを生成する)。一方、ステップS408において、フォームデータに対応するフォームラスタがフォームラスタ格納部に登録済みであると判断された場合は、続く印刷データを解析するためにステップS401へ戻る。

【0044】ステップS403において、印刷データが排紙命令(あるいは排紙命令に相当する命令)であった場合には、ステップS407にて描画オブジェクト格納部205aへセットされたオーバーレイデータ有無情報を調べ(ステップS412)、フォームオーバーレイすべきフォームデータが存在する場合は、フォームラスタ描画部203eが、前記ステップS407～S410にて生成済みの1バンド分の符合化フォームラスタをフォームラスタ格納部203dより読み出し、復号化した後、バンドラスタ205cへ描画する(ステップS413)。続くステップS414にて、フォームに含まれた(ステップS409にて生成した)フォントオブジェクトを1バンド分だけ描画オブジェクト格納部より読み出し、順時バンドラスタへ描画する。最後に通常ページのオブジェクトをバンドラスタへ描画し1バンド分の描画処理を終了する(ステップS415)。1ページ分の描画が終了するまで、上記ステップS412～S415を全バンド繰り返し(ステップS416)、全バンド終了したら、続くページの印刷データを処理するためにステップS401へ戻る。

【0045】なお、ここではフォーム内のフォントオブジェクトの描画ステップ(ステップS414)と通常ページオブジェクトの描画ステップ(ステップS415)とを分けて説明したが、図5の説明で後述するように、一連の描画情報を順次読み出せば、自動的にステップS

414, S415の各処理が行なわれるように構成されている。

【0046】図5は、本実施の形態における、通常ページの描画オブジェクト格納部205a内の1ページ分を示すメモリマップ(図5(a))および、構成の概略を示すブロック図(図5(b), 5(c))である。図5(a)において、ヘッダ部501は、各ページの種々の情報を格納し、電源投入時からカウントされるページ番号、印字解像度、用紙サイズ、バンドラスタ数、該ページに属する描画オブジェクトの総容量、描画済み/シ

ップ中などのページの状態を示す情報、および描画すべきオーバーレイデータの有無情報502等から構成されている。バンドテーブル503は、前記バンドラスタの数だけ用意され、各バンド内に印字されるべき個々の描画オブジェクトに対応した描画情報504がリンクされる(図5(b))。個々の描画情報、描画オブジェクトはそれぞれ、描画情報格納部504および描画オブジェクト記憶部505に生成・確保される。

【0047】図5(c)は、1つの描画情報の構成を示す概略図であり、描画オブジェクトを描画するための情報として、以下の情報を含むよう構成されている。すなわち、描画オブジェクトの種類、バンド内の印字位置(左上端からのビットオフセット値)、描画オブジェクトの描画高さ、バンドラスタとの描画論理、対応する描画オブジェクトの先頭アドレス、描画オブジェクトがバンド途中から描画される場合に該描画オブジェクトを読み飛ばすオフセット量(ライン数)、続く描画情報の先頭アドレス(リンク最後はNULLとする)等が含まれる(各情報は、少なくとも指定用紙を指定解像度で表現可能なだけのビット数があればよい)。ここで、上記描画オブジェクトの種類は、2値ビットマップ、多値イメージ、矩形、ランレングス形式にエンコードされた図形データ、圧縮フォントパターン、ラインから構成されているものとするが、本発明においては、プリンタエンジンへ送出するためのビットイメージを生成可能な中間形式として表現されていれば、描画オブジェクトの形式そのものについては特に規定しない。

【0048】図4のステップS415において、バンド毎に通常ページの描画オブジェクトや、フォームに含まれるフォントオブジェクトを描画する処理は、上記バンドテーブルからリンクされている描画情報を順次読み出し、各描画情報内の印字位置情報等に従って描画オブジェクトを描画する。フォーム内フォントオブジェクトは通常ページオブジェクトよりも先に生成されるため、描画情報も先にリンクされ、結果として先に描画される。また、図4のステップS410において、フォームラスタを生成する場合は、上記と全く同様の構成とすることで、同じ描画オブジェクト描画部203cを用いた生成が可能となる。ただし、フォームデータ内のフォントパターン生成時に、各フォントパターンに対応する描画情

報はリンクしないことで、フォームラスタ内にフォントパターンは描画されない。

【0049】図6は、フォームメモリ管理部203dによって管理される、フォームラスタ格納部205dのメモリ構成の概略を示すメモリ構成図である。同図において、フォーム毎の管理テーブル601は、各フォームを描画・識別するために必要な情報から構成される。管理テーブル601のための記憶領域は、予め固定領域に確保しておいても構わないし、可能なだけ動的に確保されるよう構成されていても構わない。管理テーブル601内には、各フォームを識別するためのフォームID(ホストコンピュータと同一番号で管理する)、印字解像度、分割バンド総数、用紙の向き、バンド幅、フォームラスタバンドテーブル602の先頭アドレスが含まれる。フォームラスタバンドテーブル602は、図4のステップS410において分割された、バンド状のフォームラスタ単位の情報を格納するテーブルであって、管理テーブル601内のバンド数に応じた数だけ確保され、各フォームラスタの先頭アドレスと、各バンドの高さを含む。なお、最下端のバンド以外の高さが一定であれば、前記一定値と最下端バンド高さをフォーム管理テーブル601内に保持しても良い。また、管理テーブル601に格納する情報のうち、解像度・用紙の向き・バンド幅は、印刷データやパネル規定値を元にセットされ、フォームIDはオーバーレイ登録命令のパラメータとしてホスト上のプリンタドライバなどにより指示される。

【0050】図7は、フォームメモリ管理部203dが、要求されたフォームデータに対応するフォームラスタがフォームラスタ格納部に登録済みかどうかを調べる手順について示したフローチャートである。すなわち、図4のステップS408の詳細を示した手順である。まず、同図のステップS701において、フォームラスタ格納部205d内のフォーム管理テーブル601から、第1のフォームラスタ情報を読み出し、フォームオーバーレイ実行命令で指定されたフォーム情報の各々と一致するかどうかを調べる(ステップS702~S706)。すなわち、フォームIDを比較し(ステップS702)、同じであれば次のフォーム情報を比較し、同じでなければ、次のフォームラスタ情報を読み出すために、ステップS708へ進む。以下同様に、印字解像度(ステップS703)、縦方向/横方向の用紙搬送方向がサポートされている用紙サイズ場合は用紙の向き(ステップS704)、カラープリンタの場合はカラーモード(ステップS705)、ページ拡大/縮小指定がなされているかどうか(ステップS706)をそれぞれ調べ、全ての条件が一致した場合には、ステップS707にてフォーム番号をフォームラスタ有無情報として記憶し、指定フォームオーバーレイに対応するフォームラスタが登録済みであると判断する。

【0051】一方、ステップS702~S706のいず

れかの情報が一致しなかった場合は、ステップS708にて全フォームラスタの検索が終了したかどうかを調べ、終了していなければステップS701へ戻り、全フォームラスタの検索が終了した場合は、フォームラスタは未登録であると判断される（ステップS709）。

＜ROMデータからのフォームラスタの生成＞図8は、フラッシュROMやハードディスク等の書き込み可能な不揮発性デバイス（以下、このデバイスをフラッシュROMとして説明する）内にページ記述言語形式のフォームデータが格納済みの場合の、フォーム画像生成手順を示すフローチャートである。すなわち、フラッシュROMによりフォームデータが提供されている場合には、図4の手順のみならず、図8の手順によってもフォームラスタが生成され、保持される。

【0052】本フローチャートの処理は、画像出力装置に電源が投入された直後、印刷データを受信する前や、処理すべき印刷データを受信していない空き時間等に実行しても構わないし、あるいは、通常ページの印刷データを受信する前に、フォームラスタ生成命令をホストコンピュータより受信し、実行するよう構成されていても構わない。

【0053】同図において、まずステップS801にて、フラッシュROMが装着されているかどうかを調べ、未実装であった場合は処理を終了する。一方、実装されている場合は、ステップS802にて、パネルメニュー等によって設定された印刷条件（印字解像度や、用紙搬送方向、カラーモード等）を読み出し、続くステップS803以降にて、フラッシュROMに記憶されたフォームデータからフォームラスタを生成し、フォームラスタ格納部へ登録する。まず、フラッシュROM内のフォームデータが既にフォームラスタ格納部に登録されているかどうかを調べ（ステップS803）、未登録であればフォームデータを読み出し（ステップS804）、フォントパターンを含む1ページ分の描画オブジェクトを生成し（ステップS805）、各フォントパターンはフォントキャッシュ手段へ登録する。

【0054】続くステップS806にて、印刷データを受信したかどうかを調べ、受信バッファに印刷データが存在する場合は、フォームデータに対応する生成済みの描画オブジェクトを削除し（ステップS814）、作業メモリとして使用していたワークメモリを解放した後、フォーム画像生成処理を終了する。ここで、フォーム画像の格納途中であれば、途中まで格納したフォーム画像も解放する。なお、本実施の形態では、ステップS806においてのみ、受信データの存在をチェックしたが、これに限らず、比較的处理時間が多くかかるステップの前後や、タイマ等によって定期的に印刷データを確認しても良く、こうすることによって、例えばフォームを用いない通常データの印刷処理を妨げることがなくなる。

【0055】続くステップS807にて、フォントオブ

ジェクト以外の前記描画オブジェクトを用いて1バンド分のフォーム画像をバンドバッファへ描画した後、ステップS808で前記フォーム画像を符合化する。続くステップS809において、フォーム画像格納部に登録に必要なメモリ容量があるかを調べ、空さメモリ容量が不足した場合は、ステップS810にてエラー表示を行なった後、処理を終了する。一方、登録可能であれば、符合化済みフォーム画像を格納する（ステップS811）。上記ステップS806～S811の処理を1ページ分全て終了するまで繰り返し（ステップS812）、最後にステップS813にて、フォントキャッシュ以外の不要となった描画オブジェクトを削除し、1フォームの処理を終了する。なお、本実施の形態では、メモリ不足時にエラー表示を行ない処理を中断したが、パネル設定等によって各フォームの優先順位を指定したり、使用頻度をフォームデータと共に記憶するよう構成し、優先順位の低い順、あるいは使用頻度の低い順にフォーム画像格納領域から削除して、該フォーム画像格納の処理を続行しても構わない。

【0056】以上の手順により、フォームデータ登録により登録された、ページ記述言語で記述したフォームデータ、あるいはあらかじめROMにより提供されている、ページ記述言語で記述したフォームデータを、フォームデータ生成命令により中間データによる描画オブジェクトに変換し、さらにそのフォームのラスタデータを生成して符号化し、それを記憶しておく。排紙命令を受けてあるページを排紙中に、そのページにオーバーレイがあることが判明したなら、記憶されている符号化されたフォームラスタデータを復号してバンドメモリに展開し、その上にフォーム以外の通常ページオブジェクトを展開して印刷する。

【0057】また、フォームデータと合わせて、そのフォームを構成する文字データについては、ビットマップデータに展開された形式でフォントキャッシュに保持されるため、フォームデータについては、それに含まれる文字も含めてキャッシュしておき、再利用することが可能となる。

【0058】このようにすることで、いったん登録されたフォームデータは符号化されたラスタデータとして記憶されるため、ページ記述言語から展開する必要がなく、迅速にバンドメモリに展開できる。

【0059】なお、フォームデータがページ記述言語でなく、イメージデータで送信されてくるような場合には、そのラスタデータへの展開処理が省かれるだけであり、それをそのまま符号化して格納すればよい。

【0060】[第2の実施の形態] 第1の実施の形態では、フォームデータに含まれる文字データと、通常ページで使用される文字データとは、区別なくフォントキャッシュ手段に登録されていたため、メモリ不足等により、フォーム内で使用されるフォントキャッシュが削除

10

20

30

40

50

されてしまう可能性もあった。

【0061】そのため、本実施の形態では、フォントキャッシュ手段に登録されたフォントパターンのうち、フォームデータに含まれる文字データから生成・登録されたフォントパターンに対しては、通常ページに含まれる文字データよりも優先順位を高く設定することで、フォームデータ内で使用しているフォントキャッシュは通常ページ内でのみ使用されている文字データに比べ削除されにくいよう構成する。

【0062】図9は、本発明の第2の実施の形態を示す画像出力装置の基本構成を説明するブロック図であり、第1実施の形態と同様の構成については同一番号をつけ説明を省略する。同図において、優先順位設定・判定部901は、フォントキャッシュ管理部203gへフォントパターンを登録する際に優先順位を設定し、フォントパターン削除時にどのフォントパターンから削除するかを管理する。これ以外の構成は図2と同様である。＜優先順位の付されたフォントキャッシュの構造＞図10は、本実施の形態におけるフォントキャッシュ格納部に格納されるフォントキャッシュの構成を示す概略図である。

【0063】本実施の形態においては、各フォントパターン1006は、文字コードを検索キーとしてフォント情報管理テーブル501（以下、管理テーブル）からインデックス1002及び1003をたどって検索できるようになっている。ここで、外部から与えられる文字コードは1バイトまたは2バイトであり、インデックスは文字コードの上位バイト用（1002）と下位バイト用（1003）の2段階の深さを持つ。フォントパターンをフォントキャッシュへ登録する時には、文字コードの上位バイト／下位バイトのそれぞれに対応する位置に、続く下位インデックスあるいはフォントキャッシュ実体へのポインタを書き込む（インデックス内の初期値はNULLとしておく）。一方、フォントキャッシュ検索時には、まず前記文字コードの上位バイトに対応するインデックスの位置に下位バイト用インデックスのポインタがセットされているかどうかを調べ、セットされている場合は、下位インデックスを文字コードの下位バイトで検索し、同様にフォントキャッシュの実体へのポインタを得る。なお、シングルバイト系のコード体系では、上位バイト用インデックスはコード=0x00のメンバのみから構成される。

【0064】同図において、管理テーブル1001には、各フォントを識別するための情報として以下の情報がフォント毎に格納されている。すなわち、書体名称、文字コードと実パターンの割り付けを示すグラフィックセット、文字サイズ、スケーラブルフォントか否かを示すフラグ、文字の線の太さを示すストロークウェイト（不図示）、固定ピッチかプロポーショナルピッチかを示すフラグ（不図示）、縦書き／横書きを示すフラグ

（不図示）の他、フォントパターンを検索するための上位インデックス1002へのポインタなどが格納されており、各管理テーブル毎にフォント番号が割り振られている（図10中では1番と2番）。同図では、フォント番号=1番の管理テーブルには、ダブルバイト系フォントが割り当てられており文字コード=0x2422および0x3021に対応するフォントパターンがリンクされている場合を示している。

【0065】フォントキャッシュ1004は、ヘッダ情報部1005とフォントパターン1006から構成されている。ヘッダ情報部1005には、フォントパターンのドット幅w、ドット高さhや、回転角度、フォントパターンの先頭アドレス、ベースラインからのオフセット値やピッチ情報などの描画に必要な情報に加え、拡大やボールド修飾の有無といった修飾情報などのフォントキャッシュの検索判定キーとなるフォント情報がセットされている。また、文字パターンが圧縮されて格納されている場合は、上記情報に加えて、圧縮されていることを示す符号化フラグも持つ。

【0066】さらに本発明にかかるフォントキャッシュでは、該フォントキャッシュが定型フォーム部内で使用されたかどうかを示す優先順位フラグ1005aを持ち、フォントキャッシュ作成時に、フォームデータ解析中であれば、すなわちフォームデータに含まれる文字データであれば、優先順位フラグ1005aを1にセットし、そうでなければ（通常ページ解析中に作成されたフォントキャッシュであれば）0をセットしておく。また、フォントキャッシュを検索して求めるフォントパターンが見つかった（キャッシュにヒットした）際に、フォームデータ解析中であり、かつ優先順位フラグ1005aが0にセットされている場合は、該優先順位フラグを2にセットし直す。なお、優先順位フラグ1005aが2にセットされている場合は、フォーム内と通常ページの両方から使用されたフォントキャッシュであることを示す。

【0067】一方、フォントキャッシュにヒットした際に、通常ページデータ解析中であり、かつ優先順位フラグが1にセットされている場合も、該優先順位フラグを2にセットし直し、前記同様にフォーム内と通常ページの両方から使用されたフォントキャッシュであることを示す。

【0068】なお、図10では、フォントパターン1006はRAM205内に生成された文字パターンとして説明したが、ドットフォント等のようにフォントROM204内に予めフォントパターンを持つ場合は、RAM205内にフォントパターンは存在しなくても構わない。この場合、ヘッダ情報部1005内の前記フォントパターンの先頭アドレスは、フォントROM204内のフォントパターン先頭のアドレスがセットされていることになる。

【0069】上記に説明した構成のフォントキャッシュにおいて、フォントキャッシュ削除時（メモリ不足時等）には、優先順位フラグが0→1→2の順序で削除することで、通常ページでのみ使用されているフォントキャッシュが第1に削除され、続いてフォーム内のみで使用されているフォントキャッシュ、最後にフォーム内と通常ページの両方で使用されているフォントキャッシュが削除されることになる。このような手順により、定型フォーム部で使用されるフォントキャッシュが削除されにくくなり、特に複数ページでフォームを使用する場合にフォントキャッシュのヒット率を高めることが可能となる。

【0070】図11は上述したフォントの展開手順を示すフローチャートである。フォントを展開するためのメモリ容量が十分でない場合には（ステップS111-YES）、削除しきい値を0に設定する（ステップS112）。次にその削除しきい値以下の優先順位を有するフォントキャッシュを、必要なメモリ容量が確保できるだけの量か、あるいはすべての何れか少ないほうだけ削除する（ステップS113）。それにより必要なメモリが確保できたか判定し（ステップS114）、確保できればそのメモリを利用してフォントを展開する（ステップS115）。また、メモリが十分でなければ、現在の削除しきい値が2であるか判定し（ステップS116）、2であればすべてのフォントキャッシュを削除してもまだメモリ容量が不足しているということになり、これはエラーとする。また、削除しきい値が0または1であれば、現在の削除しきい値を一つ上げて、次の優先順位のフォントキャッシュを削除する。

【0071】このようにして、フォーム部で使用されるフォントキャッシュは削除されにくくなり、特に複数ページでフォームを使用する場合にフォントキャッシュのヒット率を高められ、印刷処理の効率を向上させることができる。

【0072】なお、優先順位1あるいは2のフォントキャッシュが含まれていることをフォントの種類毎に示すために、スケーラブルフラグ欄等に、高優先度の文字が含まれていることを表示するフラグを設けてもよい。この場合には、優先順位フラグが1または2にセットされるときに、同時にこの高優先度（優先順位1または2）の文字が含まれていることを示すフラグもセットされる。また、フォントキャッシュを削除する場合には、まずフォントの種類毎に、そのフォントに高優先度の文字が含まれているか判定し、含まれていなければそのフォントから削除する。含まれていることがわかれば、他のフォントを調べて同様の処理をくり返す。

【0073】なお、上記の優先順位以外に、修飾の有無、フォントパターンサイズ、JISの第2水準/第1水準漢字/第1水準非漢字といった情報によってもフォントキャッシュを削除する優先順位を設定しても構わな

い。

【0074】

【他の実施形態】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0075】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを

読出し実行することによっても達成される。

【0076】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0077】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0078】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0079】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0080】

【発明の効果】以上説明したように、本発明によれば、1ページ分のフォームデータを符号化した画像データとして保持し、フォームオーバーレイ実行命令受信時に、対応する前記フォーム画像を復号化することで、フォームデータ内容に関わらず高速に描画処理を終了することが可能となる。特に、同一内容のフォームデータに対して通常ページが毎ページ異なるような複数ページの印刷命令においては、2ページ目以降のフォームデータの生成に要する時間を小さく抑えることができる。

【0081】特に、文字データを含まないフォーム画像を符号化するように構成したため、符号化手段の符号化率（圧縮率）を高め、メモリの使用効率を向上させると

同時に、符合化・複合化に要する処理時間を短縮できる。

【0082】また、同一の描画オブジェクトをページ毎に個別に持つ必要がなくなるため、排紙待ちなどの処理速度の低下や、描画に要するメモリを低く抑えることが可能となる。

【0083】また、フォームデータが不揮発性デバイスに記憶されている場合は、電源投入時やホストから印刷すべきデータを受信していない空き時間を用いて、フォームデータに対応するフォーム画像を生成、符合化して

おくことで、1 ページ目のファーストプリントをも高速に出力することができる。

【0084】また、1 ページをバンド状に分割した単位で前記フォーム画像を生成・符合化・保持しておき、フォームオーバーレイ指定時に前記フォーム画像を順次復号化・読み出してバンドバッファへ描画するように構成したことで、復号化の際に前処理や読み飛ばしの処理が不要となり、通常ページデータをバンディング方法によって描画を行なう画像出力装置に対して簡単に適用可能となる。

【0085】さらに、フォントキャッシュに登録されたフォントパターンのうち、フォームデータに含まれるフォントパターンに対しては、フォーム以外の通常ページに含まれる文字データよりも優先順位を高く設定して削除されにくくすることで、フォームデータに含まれるフォントキャッシュがヒットし易くなり、複数ページのフォーム印刷における処理速度を向上することができる。

【0086】

【図面の簡単な説明】

【図1】本発明を適用可能なレーザービームプリンタの断面図である。

【図2】本発明の第1の実施の形態に示す画像出力装置のブロック図である。

【図3】第1の実施の形態に示す画像出力装置における、フォームオーバーレイ登録処理とフォームラスタ登録処理、並びにフォームオーバーレイ実行手順の概略を示すフローチャートである。

【図4】第1の実施の形態に示す画像出力装置における、フォームオーバーレイ登録、フォームラスタ登録、フォームオーバーレイ実行処理のフローチャートである。

【図5】第1の実施の形態に示す画像出力装置の、描画オブジェクト格納部のブロック図である。

【図6】第1の実施の形態に示す画像出力装置の、フォ

ームラスタ格納部のブロック図である。

【図7】第1の実施の形態に示す画像出力装置の、フォーム画像検索手順のフローチャートである。

【図8】不揮発性メモリにフォームデータが格納されている場合の、第1の実施の形態に示す画像出力装置のフォームラスタ生成・登録の手順のフローチャートである。

【図9】本発明の第2の実施の形態に示す画像出力装置のブロック図である。

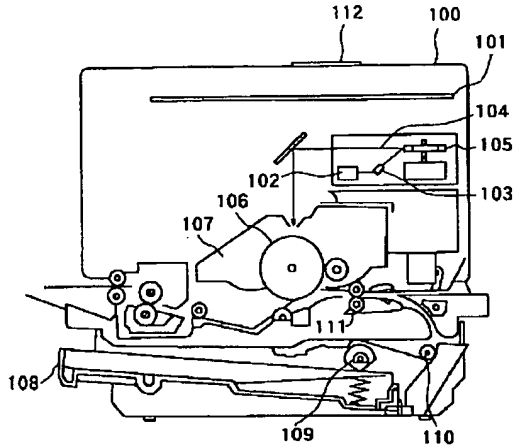
【図10】第2の実施の形態に示す画像出力装置における、フォントキャッシュ格納部のブロック図である。

【図11】第2の実施の形態に示す画像出力装置の、フォント展開処理のフローチャートである。

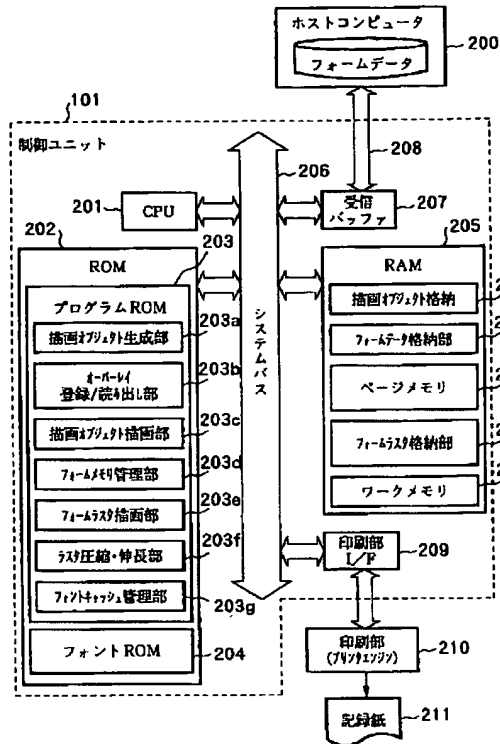
【符号の説明】

- 100 印刷装置
- 201 CPU
- 203 プログラムROM
- 203a 描画オブジェクト生成部
- 203b オーバーレイ登録／読み出し部
- 203d フォームメモリ管理部
- 203e フォームラスタ描画部
- 203f ラスタ符合化・復号化部
- 203g フォントキャッシュ管理部
- 205 RAM
- 205a 描画オブジェクト格納メモリ
- 205b フォームデータ格納部
- 205c ページメモリ（バンドラスタ）
- 205d フォームラスタ格納メモリ
- 207 受信バッファ
- 501 ページ情報ヘッダ部
- 502 オーバーレイ有無情報
- 503 バンドテーブル
- 504 描画情報格納メモリ
- 505 描画オブジェクト記憶部
- 601 フォームラスタ管理テーブル
- 602 フォームバンドテーブル
- 603 符合化フォームラスタ
- 901 優先順位設定・判定部
- 1001 フォント情報管理テーブル
- 1002・1003 インデックス
- 1004 フォントキャッシュ
- 1005 フォントキャッシュヘッダ情報部
- 1006 フォントパターン

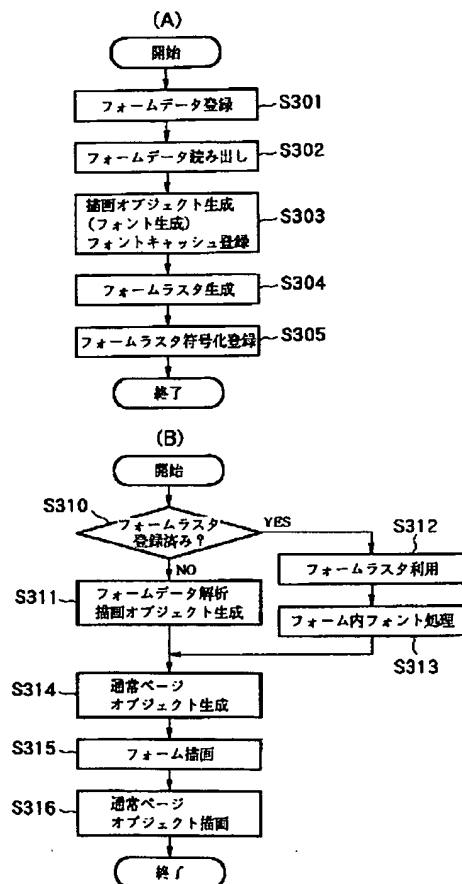
【図1】



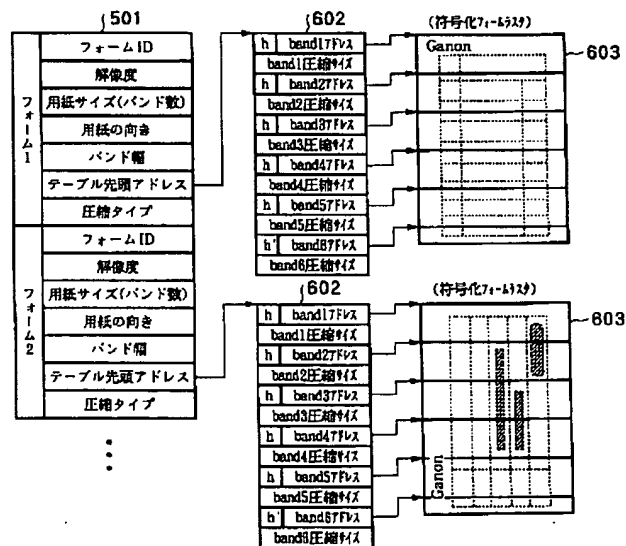
【図2】



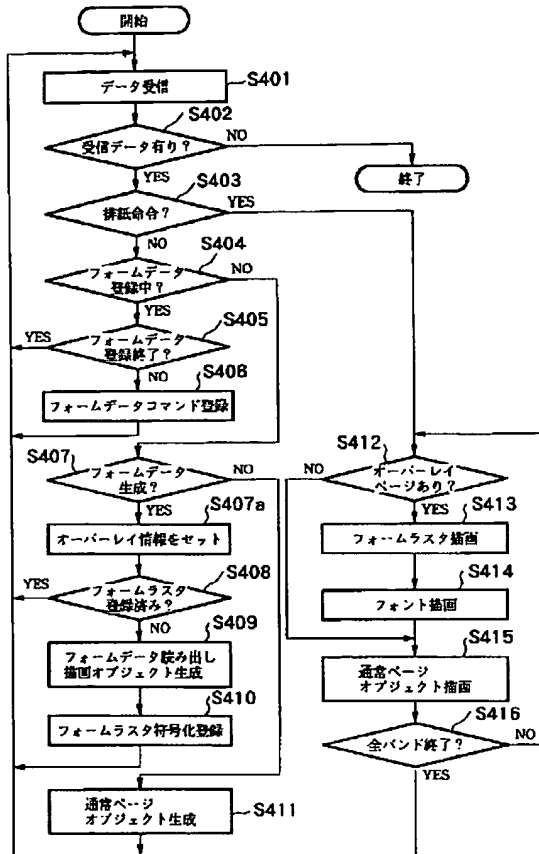
【図3】



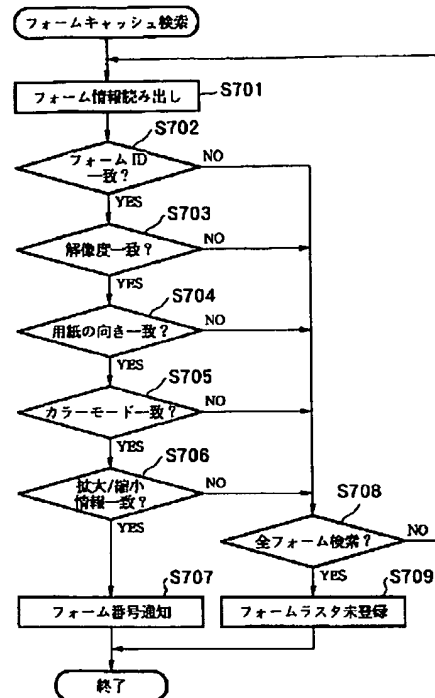
【図6】



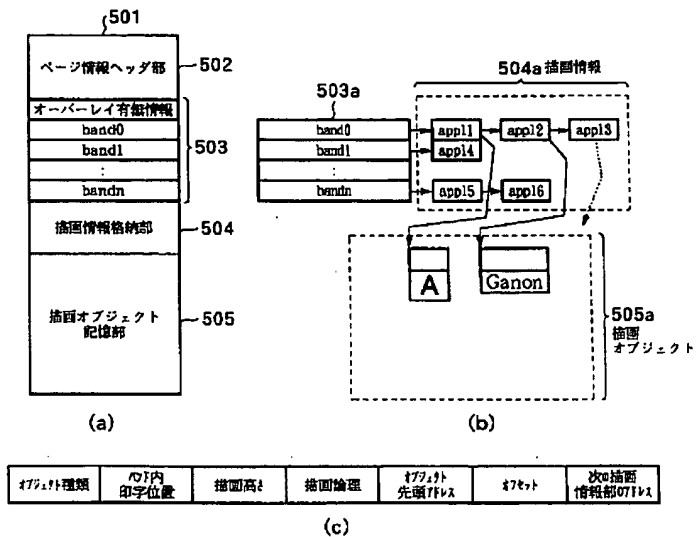
【図4】



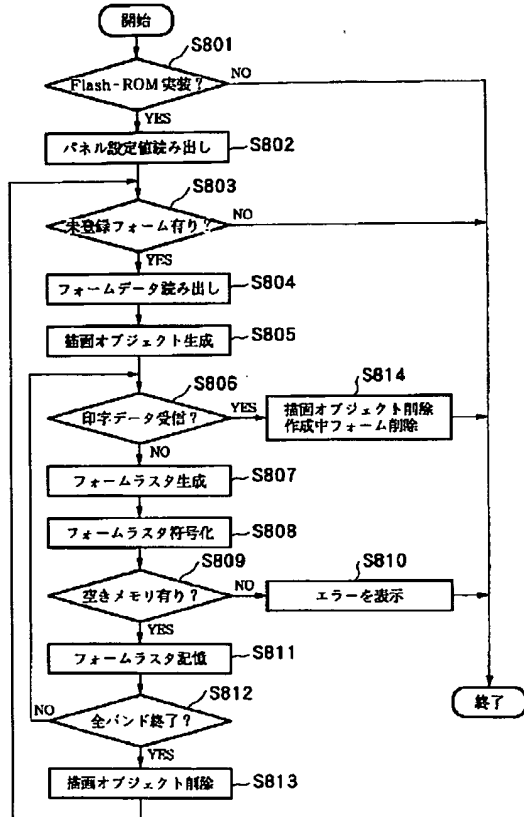
【図7】



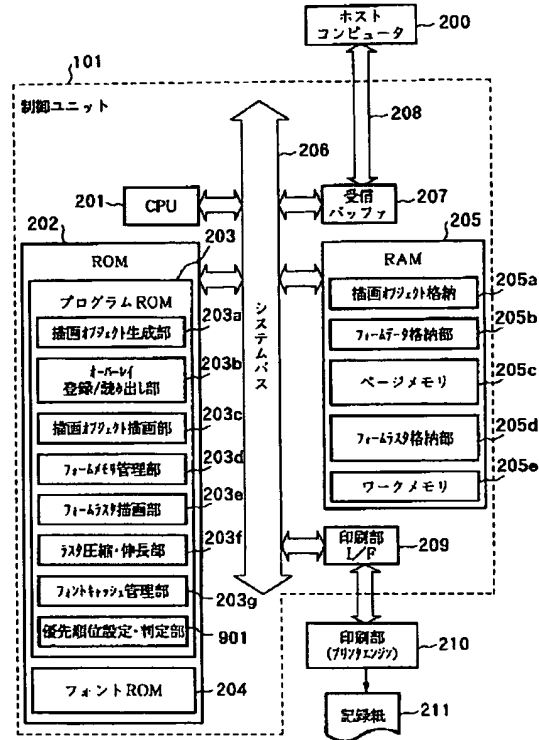
【図5】



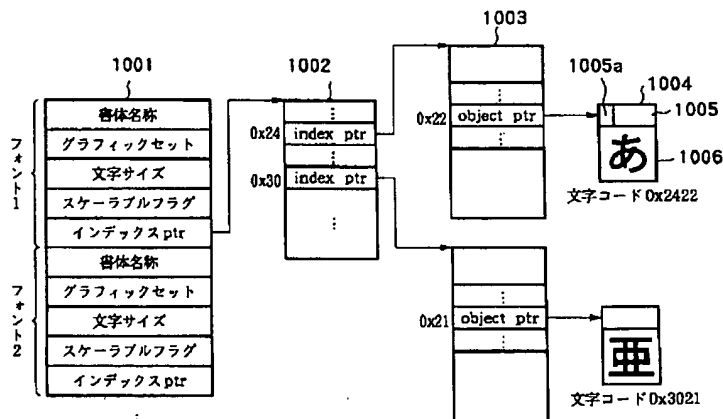
【図8】



【図9】



【図10】



【図11】

